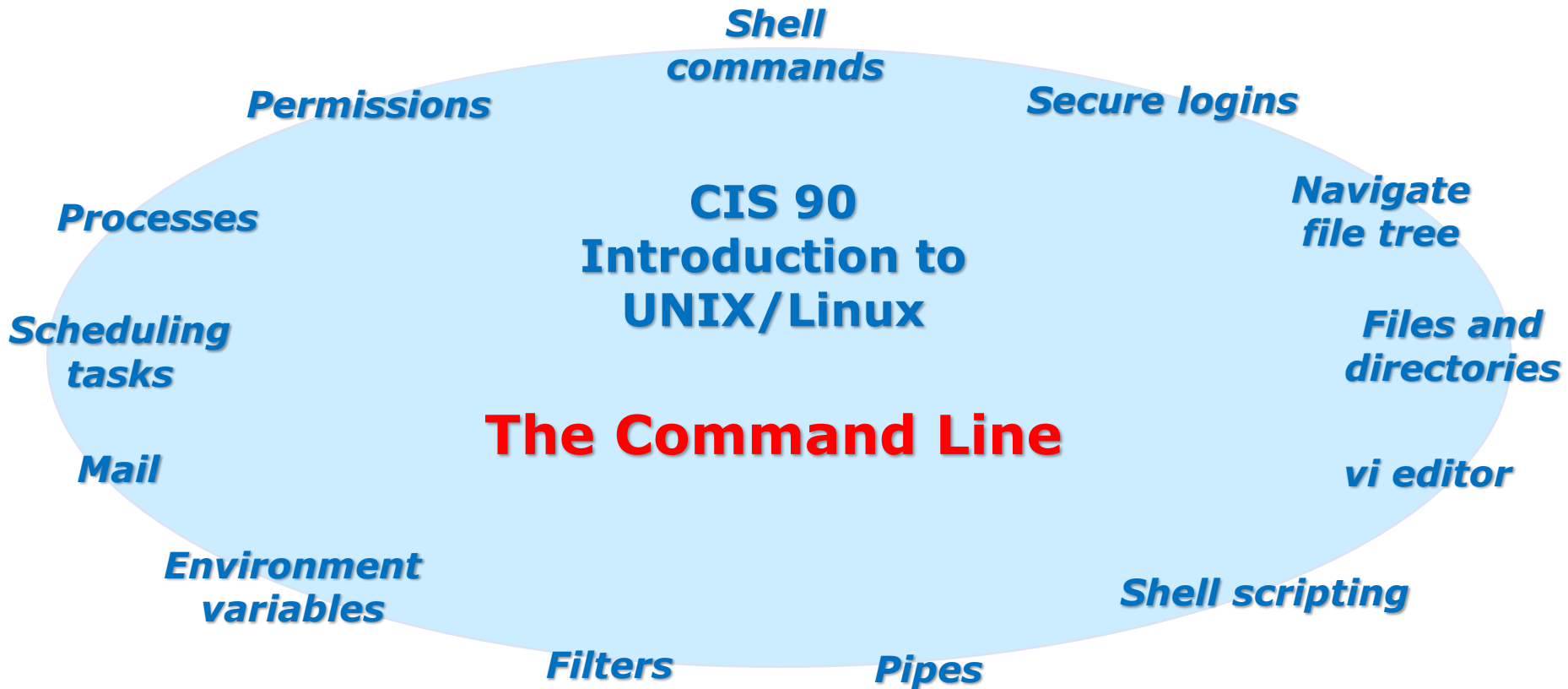




## Rich's lesson module checklist

*Last updated 04/16/2019*

- ☐ Zoom recording named and published for previous lesson
- ☐ Slides and lab posted
- ☐ Print out agenda slide and annotate page numbers
- ☐ Flash cards
- ☐ 1<sup>st</sup> minute quiz
- ☐ Calendar page updated
- ☐ Lab 9 tested and uploaded
- ☐ Test 2 stats run
- ☐ Test and schedule langs file email for Lab 9 ready (at end of class)
- ☐ Schedule lock/unlock turnin directory (scripts/schedule-submit-locks)
- ☐ Apache configured for student websites
  - ☐ /etc/httpd/conf.d/userdir.conf
    - ☐ UserDir directive
  - ☐ systemctl restart httpd
  - ☐ setsebool -P httpd\_enable\_homedirs true
  - ☐ chcon -R -t httpd\_sys\_content\_t cis90\_html
- ☐ Swap all egg & treat slides in shell six steps
- ☐ Backup slides, CCC info, handouts on flash drive
- ☐ Spare 9v battery for mic
- ☐ Key card for classroom door
- ☐ <https://zoom.us>
  - ☐ Putty, slides, Chrome
  - ☐ Enable/Disable attendee sharing
    - ^ > Advanced Sharing Options > Only Host
  - ☐ Enable/Disable attended annotations
    - Share > More > Disable Attendee Sharing



### **Student Learner Outcomes**

1. Navigate and manage the UNIX/Linux file system by viewing, copying, moving, renaming, creating, and removing files and directories.
2. Use the UNIX features of file redirection and pipelines to control the flow of data to and from various commands.
3. With the aid of online manual pages, execute UNIX system commands from either a keyboard or a shell script using correct command syntax.

# Introductions and Credits



Jim Griffin

- Created this Linux course
- Created Opus and the CIS VLab
- Jim's site: <https://web.archive.org/web/20140209023942/http://cabrillo.edu/~jgriffin/>



Rich Simms

- HP Alumnus
- Started teaching this course in 2008 when Jim went on sabbatical
- Rich's site: <http://simms-teach.com>

And thanks to:

- John Govsky for many teaching best practices: e.g. the First Minute quizzes, the online forum, and the point grading system. John's site: <http://teacherjohn.com/>
- Jaclyn Kostner for many webinar best practices: e.g. mug shot page.



## Student checklist - Before class starts

The screenshot shows a web browser window with the URL [simms-teach.com/cis90calendar.php](http://simms-teach.com/cis90calendar.php). The page title is "Rich's Cabrillo College CIS Classes CIS 90 Calendar". On the left sidebar, there are links for "CIS 90", "CIS 90A", "CIS 90B", "CIS 90C", "CIS 90D", "CIS 90E", "CIS 90F", "CIS 90G", "CIS 90H", "CIS 90I", "CIS 90J", "CIS 90K", "CIS 90L", "CIS 90M", "CIS 90N", "CIS 90O", "CIS 90P", "CIS 90Q", "CIS 90R", "CIS 90S", "CIS 90T", "CIS 90U", "CIS 90V", "CIS 90W", "CIS 90X", "CIS 90Y", "CIS 90Z". The main content area shows the "CIS 90 (Fall 2014) Calendar" with tabs for "Course Details", "Genders", and "Calendar". The "Calendar" tab is selected, showing a table with columns for "Lesson", "Date", "Topics", and "Link". The table lists lessons 1 through 10. Lesson 1 is highlighted, showing details for "Clean and Linux Overview". Below the table, there are links for "Presentation slides (download)", "Supplemental", "Assignment", "Lab 1", "Enter virtual classroom", "Quiz 1", and "Comments".

Lesson	Date	Topics	Link
1	9/2	Clean and Linux Overview • Understand how the course will work • High-level overview of computers, operating systems and virtual machines • Overview of UNIX/Linux market and architecture • Using SSH for remote network access • Using terminals and the command line	
2	9/9	Introduction to Linux • Introduction to Linux • Introduction to Linux • Introduction to Linux	
3	9/16	Introduction to Linux • Introduction to Linux • Introduction to Linux • Introduction to Linux	
4	9/23	Introduction to Linux • Introduction to Linux • Introduction to Linux • Introduction to Linux	
5	9/30	Introduction to Linux • Introduction to Linux • Introduction to Linux • Introduction to Linux	
6	10/7	Introduction to Linux • Introduction to Linux • Introduction to Linux • Introduction to Linux	
7	10/14	Introduction to Linux • Introduction to Linux • Introduction to Linux • Introduction to Linux	
8	10/21	Introduction to Linux • Introduction to Linux • Introduction to Linux • Introduction to Linux	
9	10/28	Introduction to Linux • Introduction to Linux • Introduction to Linux • Introduction to Linux	
10	11/4	Introduction to Linux • Introduction to Linux • Introduction to Linux • Introduction to Linux	

1. Browse to:  
**<http://simms-teach.com>**
2. Click the **CIS 90** link.
3. Click the **Calendar** link.
4. Locate today's lesson.
5. Find the **Presentation slides** for the lesson and **download** for easier viewing.
6. Click the **Enter virtual classroom** link to join ConferZoom.
7. Log into Opus-II with Putty or ssh command.





## Student checklist - Before class starts

☐ Google

☐ ConferZoom

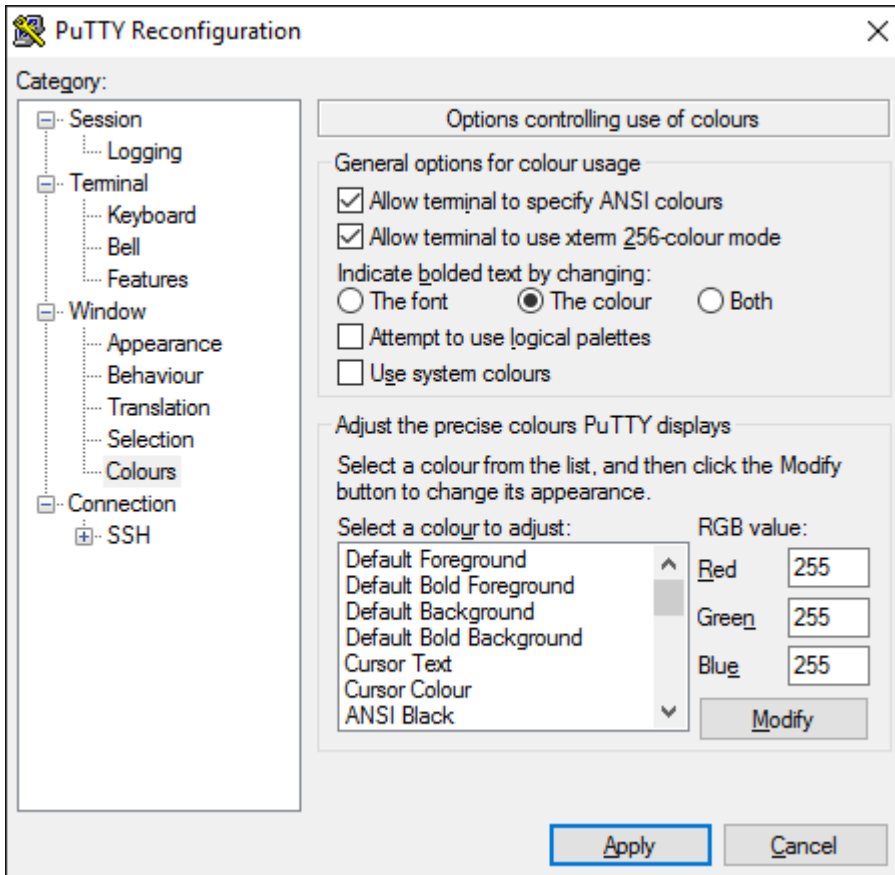
☐ Downloaded PDF of Lesson Slides. I like Foxit Reader so I can take notes using annotations.

The screenshot shows a Zoom meeting interface with several windows open. The main window displays a video of a white car with the text "Get into the car" overlaid. Other windows include the Google homepage, the Rich's Cabrillo College CIS 90 website, a PDF document titled "CIS 90 - Lesson 1", and a terminal window showing login commands. The Zoom toolbar at the bottom includes buttons for Unmute, Start Video, Invite, Participants, Share Screen, Chat, Record, and Leave Meeting.

☐ CIS 90 website Calendar page

☐ One or more login sessions to Opus-II

## Rich's checklist - Putty Colors



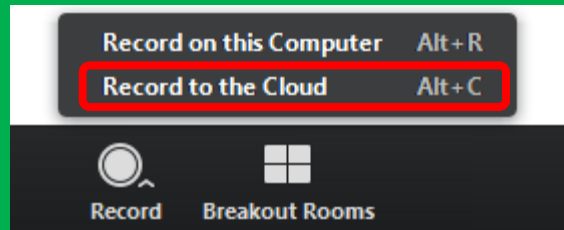
### Putty Colors

Default Foreground 255 255 255  
 Default Bold Foreground 255 255 255  
 Default Background 51 51 51  
 Default Bold Background 255 2 85  
 Cursor Text 0 0 0  
 Cursor Color 0 255 0  
 ANSI Black 77 77 77  
 ANSI Black Bold 85 85 85  
 ANSI Red 187 0 0  
 ANSI Red Bold 255 85 85  
 ANSI Green 152 251 152  
 ANSI Green Bold 85 255 85  
 ANSI Yellow 240 230 140  
 ANSI Yellow Bold 255 255 85  
 ANSI Blue 205 133 63  
 ANSI Blue Bold 135 206 235  
 ANSI Magenta 255 222 173  
 ANSI Magenta Bold 255 85 255  
 ANSI Cyan 255 160 160  
 ANSI Cyan Bold 255 215 0  
 ANSI White 245 222 179  
 ANSI White Bold 255 255 255

<http://looselytyped.blogspot.com/2013/02/zenburn-pleasant-color-scheme-for-putty.html>



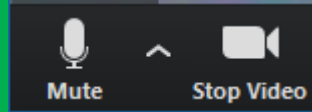
# Start



# Start Recording

Audio Check





Start Recording

# Audio & video Check



Instructor: **Rich Simms**  
Dial-in: **669-900-6833 (toll)**  
Meeting ID: **426 283 384**



Nick



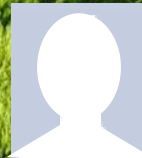
Ryan



Erik



Matt



David



Jon



Cheryl



Wais



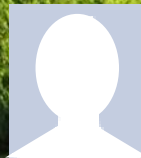
Tanisha



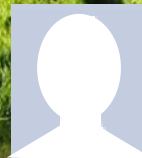
Daniel



Ohunayo



Sequoia



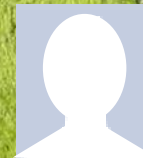
Scott



Lucky



Cole



Shane



Jim



Joseph



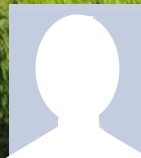
Mark



Adina



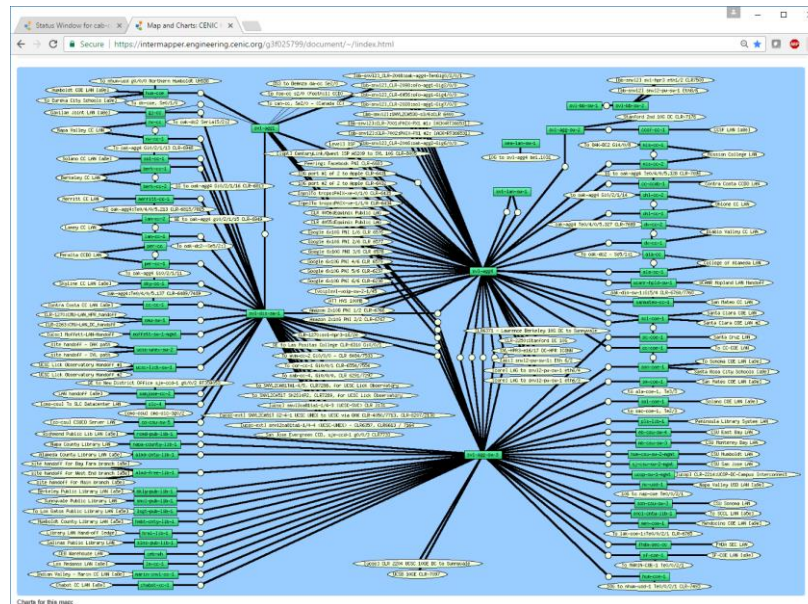
Evie



Cody



# Network Check



[https://intermapper.engineering.cenic.org/g3f025799/  
document/~!/index.html](https://intermapper.engineering.cenic.org/g3f025799/document/~!/index.html)

## First Minute Quiz

Please answer these questions **in the order** shown:

**Use ConferZoom White Board**

**email answers to: [risimms@cabrillo.edu](mailto:risimms@cabrillo.edu)**

**(answers must be emailed within the first few minutes of class for credit)**



## vi editor

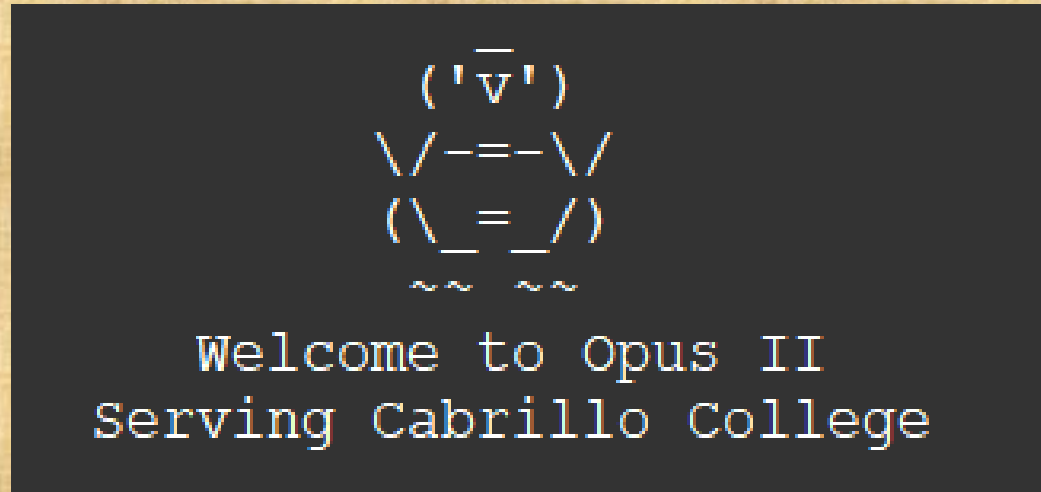
Objectives	Agenda
<ul style="list-style-type: none"><li>• Create and modify text files</li></ul>	<ul style="list-style-type: none"><li>• Quiz</li><li>• Questions</li><li>• Test 2 Post Mortem</li><li>• Housekeeping</li><li>• grep workout</li><li>• Shell Six Steps (review)</li><li>• Signals (review)</li><li>• Target Practice</li><li>• Using &amp;</li><li>• Job control (review)</li><li>• Load balancing &amp; scheduling (review)</li><li>• Text editors</li><li>• vi 101</li><li>• vi</li><li>• Tangent on spell</li><li>• Assignment</li><li>• Wrap up</li></ul>



Emmanuel Macron a fait appel à l'histoire pour réveiller l'optimisme des Français. *"Au cours de notre histoire nous avons bâti des villes, des ports, des églises (...), à chaque fois, nous les avons reconstruites", a déclaré le président solennel. Nous rebâtirons la cathédrale plus belle encore et je veux que ce soit achevé d'ici cinq années".*

-- Le Monde, le mardi 16 avril 2019

## Class Activity



If you haven't already,  
log into Opus-II

## Class Activity

3	2/19	<b>Unit 3</b> <b>Electronic Mail</b> <ul style="list-style-type: none"><li>• Guest speaker: Denise Moore on OTC (On-The-Job) training programs</li><li>• Learn how to use the LINC communication tools: write and /bin/mail</li><li>• Overview on and to and mail</li></ul> <b>Materials</b> <ul style="list-style-type: none"><li>• Presentation slides (<a href="#">download</a>)</li></ul> <b>Supplemental</b> <ul style="list-style-type: none"><li>• Howto #318: Accessing vlab (<a href="#">download</a>)</li></ul> <b>Assignment</b> <ul style="list-style-type: none"><li>• Read/skim Lesson 3 slides</li></ul>	<a href="#">Lab 2</a>
---	------	--	-----------------------

<https://simms-teach.com/cis90calendar.php>

If you haven't already,  
download the lesson slides



## Class Activity

	<ul style="list-style-type: none"><li>• <u>Read/skim Lesson 1 slides</u></li><li>• <u>Student Survey</u></li><li>• <u>Lab 1</u></li></ul>	
	<b>ConferZoom</b> <ul style="list-style-type: none"><li>• <u>Enter virtual classroom</u></li><li>• <u>Class archives</u></li></ul>	
	<b>Quiz 1</b> <b>Commanda</b> <ul style="list-style-type: none"><li>• Understand how the UNIX login operation</li></ul>	

<https://simms-teach.com/cis90calendar.php>

If you haven't already, join  
ConferZoom classroom



# Questions



# Questions?

Lesson material?

Labs? Tests?

How this course works?

- Graded work & tests in home directories
- Answers in /home/cis90/answers

*Who questions much, shall learn much, and retain much.*

- Francis Bacon

*If you don't ask, you don't get.*

- Mahatma Gandhi

Chinese  
Proverb

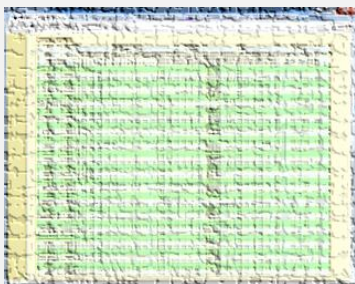
他問一個問題，五分鐘是個傻子，他不問一個問題仍然是一個傻瓜永遠。

*He who asks a question is a fool for five minutes; he who does not ask a question remains a fool forever.*

## Review your progress in the course

### Check the website Grades page

<http://simms-teach.com/cis90grades.php>



### Or check on Opus-II

**checkgrades** *codename*  
(where *codename* is your LOR codename)



Written by Jesse Warren a past CIS 90 Alumnus

- Send me your survey to get your LOR codename.
- Graded labs and tests are in your home directories.

Percentage	Total Points	Letter Grade	Pass/No Pass
90% or higher	504 or higher	A	Pass
80% to 89.9%	448 to 503	B	Pass
70% to 79.9%	392 to 447	C	Pass
60% to 69.9%	336 to 391	D	No pass
0% to 59.9%	0 to 335	F	No pass

At the end of the term I'll add up all your points and assign you a grade using this table

### Points that could have been earned:

7 quizzes: 21 points  
 7 labs: 210 points  
 2 tests: 60 points  
 2 forum quarters: 40 points  
**Total: 331 points**

## Extra Credit

### On the forum

Be sure to monitor the forum as I may post extra credit opportunities without any other notice!

### On some labs

#### Extra credit (2 points)

For a small taste of what you would learn in CIS 191 let's add a new user to your Arya VM. Once added we will see how the new account is represented in `/etc/passwd` and `/etc/shadow`.

1. Log into your Arya VM as the cis90 user. Make sure it's your VM and not someone else's.
2. Install the latest updates:  
`sudo apt-get update`  
`sudo apt-get upgrade`
3. Add a new user account for yourself. You may make whatever username you wish. The example below shows how Benji would make the same username he uses on Opus:  
`sudo useradd -G sudo -c "Benji Simms" -m -s /bin/bash simben90`

### In lesson slides (search for extra credit)



### On the website

<http://simms-teach.com/cis90grades.php>

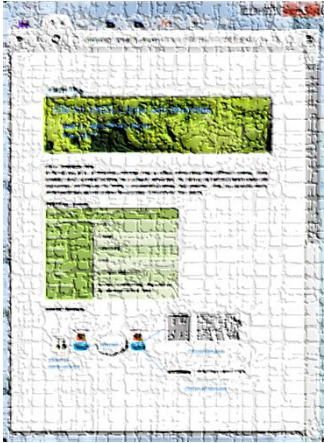
For some flexibility, personal preferences or family emergencies there is an additional 90 points available of extra credit activities.

<http://simms-teach.com/cis90extracredit.php>

- **Website content review** - The first person to email the instructor pointing out an error or typo on this website will get one point of extra credit for each unique error. The email must specify the specific document or web page, pinpoint the location of the error, and specify what the correction should be. Duplicate errors count as a single point. This does not apply to pre-published material that has been updated but not yet presented in class. (Up to 20 points total)



## Lab Assignments -- Pearls of Wisdom



- Don't wait till the last minute to start.
- Plan for things to go wrong and give yourself time to ask questions and get answers.
- The *slower* you go the *sooner* you will be finished.
- A few minutes reading the forum can save you hour(s).
- Line up materials, references, equipment and software ahead of time.
- It's best if you fully understand each step as you do it. Use Google or refer back to lesson slides to understand the commands you are using.
- Keep a growing cheat sheet of commands and examples.
- Study groups are very productive and beneficial.
- Use the forum to collaborate, ask questions, get clarifications and share tips you learned while doing a lab.
- **Late work is not accepted** so submit what you have for partial credit.

## Getting Help When Stuck on an Assignment

- Google the topic/error message.
- Search the Lesson Slides (they are PDFs) for a relevant example on how to do something.
- Check the forum. Someone else may have run into the same issue and found a way past it. If not start a new topic, explain what you are trying to do and what you have tried so far.
- Talk to a tutor/assistant at the CTC (room 1403) or CIS Lab (STEM Center).
- Come see me during my office or lab hours:

<https://www.cabrillo.edu/salsa/listing.php?staffId=1426>

**I'm in the CTC (room 1403) every Tuesday from 3:30-6:00 pm.**

- Make use of the Open Questions time at the start of every class.
- Make a cheat sheet of commands and examples so you never again get stuck on the same thing!

*CIS Labs always involve some troubleshooting!*

**Help Available!**  
In the CTC and CIS Lab

## Rich's Cabrillo College CIS Classes CIS 90 Calendar

Home

Resources

Forums

**Tutors**

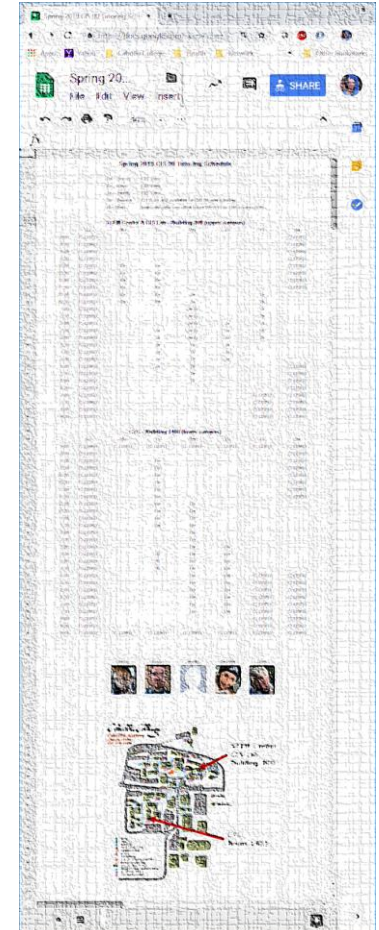
Canvas

**Cabrillo College**  
Cabrillo Gallery  
Library #1002  
831-479-6308

CIS Lab  
in STEM Center  
Building 800

*To see tutor  
schedule, click  
the Tutors link  
on the  
website.*

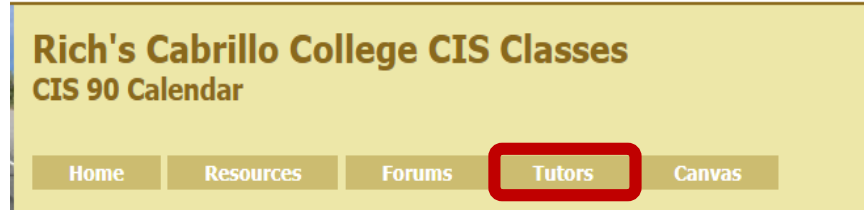
*Instructors, tutors  
and equipment are  
available for CIS  
students to work on  
assignments.*



CTC  
Room 1403



## Help Available! In the CTC and CIS Lab



*To see tutor schedule, click the  
Tutors link on the website.*



*The CIS Lab is in the STEM  
center (Building 800)*



*Room 1403 is in the  
CTC (Building 1400)*





# The slippery slope



- 1) If you didn't submit the last lab ...
- 2) If you were in class and didn't submit the last quiz ...
- 3) If you didn't send me the student survey assigned in Lesson 1 ...
- 4) If you haven't made a forum post in the last quarter of the course ...
- 5) If you had trouble doing the last test ...

*Please contact me by email, see me during  
my office hours or when I'm in the CTC*

Email: [risimms@cabrillo.edu](mailto:risimms@cabrillo.edu)



# Test 2

# Post Mortem

## Test 2 – Results

Missed Q30 = 15  
Missed Q4 = 14  
Missed Q24 = 14  
Missed Q29 = 13  
Missed Q26 = 13  
Missed Q2 = 13  
Missed Q21 = 13  
Missed Q11 = 12  
Missed Q28 = 11  
Missed Q25 = 11  
Missed Q23 = 11  
Missed Q18 = 11  
Missed Q27 = 10  
Missed Q17 = 10  
Missed Q13 = 9

Missed Q22 = 8  
Missed Q20 = 8  
Missed Q9 = 6  
Missed Q19 = 6  
Missed Q16 = 5  
Missed Q12 = 5  
Missed Q6 = 4  
Missed Q15 = 4  
Missed Q14 = 4  
Missed Q8 = 3  
Missed Q7 = 3  
Missed Q10 = 3  
Missed Q5 = 2  
Missed Q3 = 2  
Missed Q1 = 0



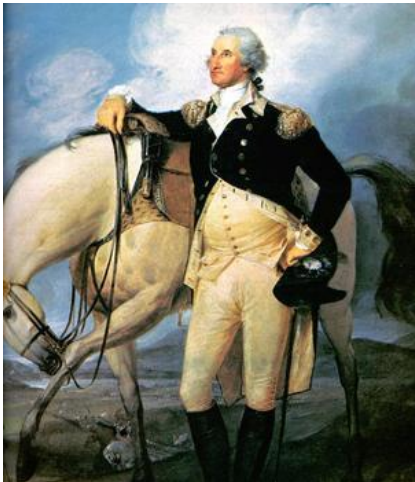
### Extra Credit

Missed Q32 = 17  
Missed Q31 = 17  
Missed Q33 = 16



Q16) There is a file in the `/etc` directory named *passwd*. This file has information on all user accounts including usernames, UIDs, first and last name, etc. What is the absolute pathname of this file?

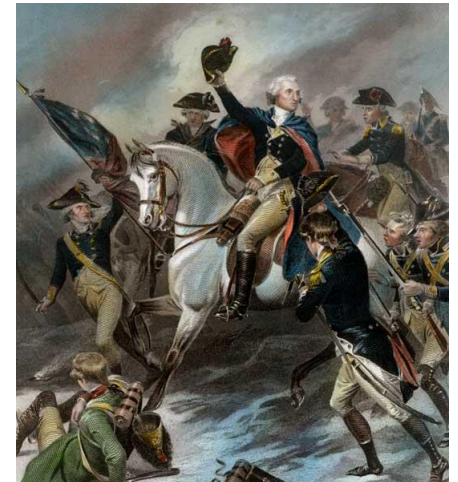
**Correct answer: `/etc/passwd`**



<http://www.sodahead.com/united-states/what-color-was-george-washingtons-white-horse/question-636725/>



<http://kids.britannica.com/comptons/art-55428/General-George-Washington-and-his-staff-welcoming-a-provision-train>



<http://www.mountvernon.org/content/revolutionary-war-princeton-white-horse>



## ConferZoom Annotations

You are viewing Rich Simms' screen

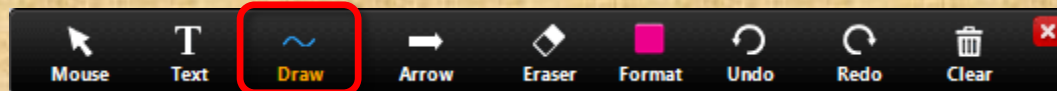
View Options ▾

Original Size

Request Remote Control

Annotate

Exit Full Screen



*Find the annotation  
drawing tool for a  
checkmark.*

*View Options > Annotate > Draw > "✓"*

Would you be interested in an online workshop on doing the Test 2 questions in 30 seconds or less?

Date: Sunday April 28th

Time: 2PM till whenever

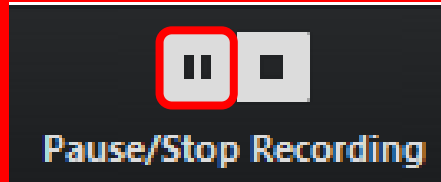
Yes	Maybe	No

*Put a checkmark indicating your interest level above*

# Housekeeping







# Pause Recording

Audio Check



# Roll Call

If you are watching the archived video please email me to let me know you were here.

[risimms@cabrillo.edu](mailto:risimms@cabrillo.edu)

# Overlap Students

Don't forget to update the Google  
Docs Log when watching the  
recording



Resume/Stop Recording

# Resume Recording

## Audio Check

# 1. Lab 8 due tonight

at 11:59pm

at> cat files.out bigshell > lab08

at> cp lab08 /home/rsimms/turnin/cis90/lab08.\$LOGNAME

at> <Ctrl-D>

*Don't wait till midnight tonight to see if this worked! Submit at an earlier time to be safe.*

# 2. A **check8** script is available for Lab 8.

# 3. Read your email on Opus to verify your Lab 8 submission was received AND that you did not submit an empty file!

# 4. Note: Lab 9 and five posts due next week.



## Heads up on Final Exam

Test #3 (final exam) is **Wednesday May 22, 7-9:50AM**

<b>Wed</b>	5/22	<p><b>Test #3 (the final exam)</b></p> <p><b>Time</b></p> <ul style="list-style-type: none"> <li>WEDNESDAY 7:00AM - 9:50AM in Room 828 or online</li> </ul> <p><b>Materials</b></p> <ul style="list-style-type: none"> <li>Presentation slides (<a href="#">download</a>)</li> <li>Test (<a href="#">canvas</a>)</li> </ul> <p><b>ConferZoom</b></p> <ul style="list-style-type: none"> <li><a href="#">Enter virtual classroom</a></li> <li><a href="#">Class archives</a></li> </ul>		<p><a href="#">5 posts</a> <a href="#">Lab X1</a> <a href="#">Lab X2</a></p>
------------	------	--	--	--

*Extra credit labs  
and final posts  
due by 11:59PM*

- All students will take the test at the same time. The test starts at **7:00AM** must be completed by **9:50AM**.
- Working and long distance students can take the test online via ConferZoom and Canvas.
- Working students will need to plan ahead to arrange time off from work for the test.
- Test #3 is **mandatory** (even if you have all the points you want)

## SPRING 2019 FINAL EXAMINATIONS SCHEDULE MAY 20 TO MAY 25

### DAYTIME FINAL SCHEDULE

**Daytime Classes:** All times in bold refer to the beginning times of classes. **MW/Daily** means Monday alone, Wednesday alone, Monday and Wednesday **or any 3** or more days in any combination. **TTH** means Tuesday alone, Thursday alone, or Tuesday and Thursday. **Classes meeting other combinations of days and/or hours not listed must have a final schedule approved by the Division Dean.**

STARTING CLASS TIME / DAY(S)	EXAM HOUR	EXAM DATE
<i>Classes starting between:</i>		
6:30 am and 8:55 am, MW/Daily	7:00 am-9:50 am	Monday, May 20
9:00 am and 10:15 am, MW/Daily	7:00 am-9:50 am	Wednesday, May 22

### CIS 90

### Introduction to UNIX/Linux

Provides a technical overview of the UNIX/Linux operating system, including hands-on experience with commands, files, and tools.

Recommended Preparation: CIS 1L or CIS 72.

Transfer Credit: Transfers to CSU;UC

Section	Days	Times	Units	Instructor	Room
1	W	9:00AM-12:05PM	3.00	R.Simms	OL
Section 1 is an ONLINE course. Meets weekly throughout the semester online during the scheduled times by remote technology with an additional 50 min arranged online lab per week. For details, see instructor's web page at <a href="http://go.cabrillo.edu/online">go.cabrillo.edu/online</a> .					
2	W	9:00AM-12:05PM	3.00	R.Simms	828
&	Arr.	Arr.		R.Simms	OL
Section 2 is a Hybrid ONLINE course. Meets weekly throughout the semester at the scheduled times with an additional 50 min online lab per week. For details, see instructor's web page at <a href="http://go.cabrillo.edu/online">go.cabrillo.edu/online</a> .					

## Your turn to grade me!

1. March 18th - NAS Office sends online SurveyMonkey survey to all CIS 90 students. Please complete it by April 17th.
2. I've added the survey link to the website Calendar page as well (see Lesson 9).
3. April 3rd - Department Chair (Mike) will visit our class to observe.

Student Evaluation of Online Instructor Richard Stevens, Spring 2018

One of the major responsibilities of the District is to provide high teaching standards among its faculty. Please take the time to evaluate the instructor of this course.

Please be thoughtful and candid in your responses. We assured that you will remain anonymous in the process and that your answers will be given the highest consideration. They will be forwarded at the end of the semester after grades have been submitted. Please do not put your name on this survey.

1. Course section:

☐ CIS 90 - Sec 1 - Introduction to LINUX/Linux

2. Is there a required text or software for your course?

☐ Yes ☐ No

3. Is there a website for your course?

☐ Yes ☐ No

4. How often do you use the online resources?

☐ Daily ☐ 2 times/week ☐ 3 times/week ☐ 4 times/week ☐ 5 times/week

5. Are the computer system requirements for the course clearly stated?

☐ Yes ☐ No

6. Is the instructor contact information clearly posted?

☐ Yes ☐ No

7. Which of the following methods of contact between the instructor and students are regularly used in the course? Check all that apply.

☐ Announcements/Email ☐ News mail with Blackboard ☐ Correspondence by private email

☐ Discussion boards ☐ Office hours ☐ Scheduled lecture/discussion ☐ Live meetings

☐ Email ☐ Scheduled lecture/discussion ☐ Live meetings

☐ Blackboard/Canvas ☐ Field trips ☐ Study and/or review sessions

☐ Telephone ☐ Study and/or review sessions

Other (please describe):

8. Rate your Online Instructor in each of the following categories:

	Excellent	Good	Acceptable	Average	Poor
A. Instructional materials such as text, papers, tests and software are appropriate and useful.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
B. Course objectives and grading criteria are specifically stated and clearly explained in course materials.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
C. Amount and type of assigned course work are appropriate.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
D. Course content is well organized.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
E. Methods of evaluation are clearly outlined and related to stated results. (If no assignment, tests, projects, writing assignments, and portfolios.)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
F. Grading is fair and impartial.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
G. Working contact with your instructor who demonstrates an enthusiasm for the subject.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
H. Working contact with your instructor who is knowledgeable and prepared.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I. Interacts with your instructor who encourages your interest, participation, and individual effort.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
J. The instructor communicates clearly.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
K. The instructor is accessible for individual communication.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
L. The instructor demonstrates respect and respect course students, including respectful consideration of any accommodations, special needs, or special requests. If noted.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
M. Numerous graded assignments in comparable size with typical comments.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
N. THE OVERALL EVALUATION OF THIS INSTRUCTOR IS:	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

9. Describe the strengths of the instructor. (Please be specific and give examples.)

10. What could the instructor do to improve teacher effectiveness or teaching style?

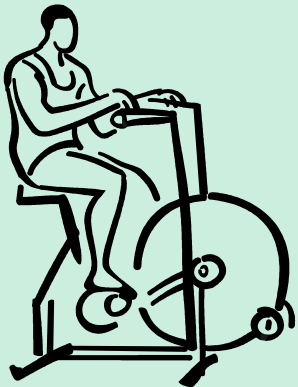
11. Would you recommend the instructor to other students and would you take another class from the instructor? Why or why not?

12. If you have additional comments, please elaborate below.

Done

See how easy it is to ENROLL ONLINE!  
Privacy & Data Policy

# grip workout







## Some perfect times to use the **grep** command:

- 1) To search through the output of a command for some text

```
command | grep "text string"
```

- 2) To search inside one or more files for some text

```
grep "text string" file1 file2 ... fileN
```

- 3) To search (recursively) inside all files in a branch of the UNIX file tree for some text

```
grep -R "text string" directory
```

## grep usage – search output of a command

Is the CUPS daemon (print service) running right now?

```
/home/cis90/simben $ ps -ef | grep cups
root      1323      1  0 Jan21 ?           00:00:24 /usr/sbin/cupsd -f
simben90  6361    3202  0 11:26 pts/1       00:00:00 grep --color=auto cups
```

*Yes it is, with PID=1323*

## grep practice

Is the cronjob daemon (**crond**) running right now?

*If so, type the crond PID into  
the chat window*

## grep usage – search output of a command

Is the Apache web server (httpd) installed?

*This shows all installed  
package names*

*This searches for package  
names containing "httpd"*

```
/home/cis90/simben $ rpm -qa | grep httpd  
httpd-tools-2.4.6-80.el7.centos.1.x86_64  
httpd-2.4.6-80.el7.centos.1.x86_64
```

*Yes, version 2.4.6 has been installed*

```
/home/cis90/simben $ httpd -v  
Server version: Apache/2.4.6 (CentOS)  
Server built: Jun 27 2018 13:48:59
```



## grep practice

Which relational DBMS (Database Management System) is installed on Opus-II?

MySQL  
PostgreSQL  
MariaDB

*Put the name and version in the chat window*

FYI, this DBMS is used by the Forum

## grep usage – search output of a command

When were the last 5 times I logged in?

```
/home/cis90/simben $ last | grep $LOGNAME | head -n5
simben90 pts/2          localhost            Sat Nov  3 16:00      still logged in
simben90 pts/6          2607:f380:80f:f8    Wed Oct 31 15:03 - 16:44  (01:41)
simben90 pts/6          2607:f380:80f:f8    Wed Oct 31 12:32 - 15:03  (02:30)
simben90 pts/2          c-73-222-184-235    Tue Oct 30 12:54 - 15:09  (02:15)
simben90 pts/0          c-73-222-184-235    Tue Oct 30 12:53 - 14:14  (01:21)
/home/cis90/simben $
```

*This scans the latest wtmp log file and lists your most recent five logins to Opus-II*

## grep practice

For the time period covered by the current wtmp log file. What was the date of your earliest login?

*Type your earliest login date into  
the chat window*

# grep usage – search output of a command

```
[rsimms@oslab ~]$ ls /bin/{bash,sh,ksh,csch,tcsch}
/bin/bash  /bin/csh  /bin/ksh  /bin/sh  /bin/tcsch
```

```
[rsimms@oslab ~]$ ksh
$ sh
sh-4.2$ csh
```

*Look familiar? (lab 8) Shows how to compare shells by size and record the biggest one in a file.*

```
[rsimms@oslab ~]$ ps -l
```

*size*

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	1201	9483	9476	0	80	0	- 28881	do_wai	pts/1	00:00:00	bash	
0	S	1201	9533	9483	0	80	0	- 29280	do_wai	pts/1	00:00:00	ksh	
0	S	1201	9557	9533	0	80	0	- 28847	do_wai	pts/1	00:00:00	sh	
0	S	1201	9561	9557	0	80	0	- 29876	sigsus	pts/1	00:00:00	csh	
0	R	1201	9771	9561	0	80	0	- 37235	-	pts/1	00:00:00	ps	

```
[rsimms@oslab ~]$ ps -l | grep csh
```

0	S	1201	9561	9557	0	80	0	- 29876	sigsus	pts/1	00:00:00	csh
---	---	------	------	------	---	----	---	---------	--------	-------	----------	-----

```
[rsimms@oslab ~]$ ps -l | grep csh > bigshell
```

```
[rsimms@oslab ~]$ cat bigshell
```

0	S	1201	9561	9557	0	80	0	- 29876	sigsus	pts/1	00:00:00	csh
---	---	------	------	------	---	----	---	---------	--------	-------	----------	-----

## grep practice

*Instructor note:*

*Login directly to simben90 (don't su)*

*Give write permission to others on Benji's terminal device: **chmod o+w \$(tty)***

- Run **bash**, **ksh**, **sh** and **cs** shells and use **ps -l** to see which is the smallest.
- Redirect the line of **ps -l** output for the smallest shell to Benji Simms's terminal: **/dev/pts/??**
- Sign it with **echo "From *first name*" > /dev/pts/??**
- Then **exit** each shell till your are back to just one bash shell running.



## grep usage – search inside files

How many CIS 90 user accounts are there?

```
/home/cis90/simben $ grep :1090: /etc/passwd | wc -l  
43
```

```
/home/cis90/simben $ grep cis90 /etc/passwd | wc -l  
43
```

```
/home/cis90/simben $ grep "^.*90" /etc/passwd | wc -l  
43
```

**FYI**  
only

*There are 43*

*The third example is a "regular expression". For more information see the Resources page of the website.*

## grep practice

How many CIS 76 accounts are there on Opus-II?

*Type the number of CIS 76 accounts  
into the chat window*

## grep usage – search inside files

Example: What is my account information in /etc/passwd?

```
/home/cis90/simben $ grep $LOGNAME /etc/passwd  
simben90:x:1000:90:Benji Simms:/home/cis90/simben:/bin/bash
```

or

```
/home/cis90/simben $ grep simben90 /etc/passwd  
simben90:x:1000:90:Benji Simms:/home/cis90/simben:/bin/bash
```

or

```
/home/cis90/simben $ cat /etc/passwd | grep $LOGNAME  
simben90:x:1000:90:Benji Simms:/home/cis90/simben:/bin/bash
```

Diagram illustrating the fields in the output of the command:

- username (points to `simben90`)
- password (just a placeholder now) (points to `x`)
- User ID (UID) (points to `1000`)
- Group ID (GID) (points to `90`)
- Comment (points to `Benji Simms`)
- Home directory (points to `/home/cis90/simben`)
- Shell (points to `/bin/bash`)

*Note the field separator used in /etc/passwd is a ":"*

## grep practice

Does your user ID in */etc/passwd* match the uid output by the **id** command?

*Type your answer (yes or no) and your uid from the **id** command into the chat window*

# grep usage – search inside files in all or part of the file tree

*All the system configuration files are in the /etc directory*

Where does the system set your "prompt" variable?

```
/home/cis90/simben $ grep -r "PS1=" /etc 2> /dev/null
/etc/bashrc: [ "$PS1" = "\s-\v\\\$ " ] && PS1="[\u@\h \W]\\$ "
/etc/bashrc: # PS1="[\u@\h:\l \W]\\$ "
```

*It is set more than once during login. We will learn in a future lesson that the one in .bash\_profile is done last and is what you end up using.*

```
/home/cis90/simben $ grep PS1= .bash_profile
PS1='$PWD $ '
```



## grep practice

Find the file in the */usr/share* branch of the file tree that contains "playing hot potato".

*Type the absolute pathname of the file  
in the chat window.*



# Shell

## Six Steps

### (REVIEW)

# Activity

*This is Benji's home directory*

```
/home/cis90/simben $ ls -F
1968.egg  class/      edits/      fun/        lab01-collection  log      redhat/
africa/   dead.letter errors      Hidden/     lab02-collection  mbox     sawyer
Apple/    debian/     etc/        HP/         lab04-mydata      misc/    slackware/
basket/   Dell/       f1.graded   island/     lesson7/          mylog    stuff
bigfile   Directory3/ f2.graded   jobs/       letter            normal   uhistory
bin/      docs/       five        L7-fun/     letter.bak        poems/
/home/cis90/simben $
```

*Benji wants to find some eggs and types this command*

```
/home/cis90/simben $ find / -name *egg 2> /dev/null
```

*Write what you think will happen  
in the chat window*

# Example Command

```
/home/cis90/simben $ find / -name *egg 2> /dev/null
```

```
/home/cis90/bomnic/1968.egg
```

```
/home/cis90/cis/1968.egg
```

```
/home/cis90/cormax/1968.egg
```

```
/home/cis90/jorwes/1968.egg
```

```
/home/cis90/bownic/1968.egg
```

```
/home/cis90/rodduk/1968.egg
```

```
/home/cis90/tosbre/1968.egg
```

```
/home/cis90/jordan/basket/1968.egg
```

```
/home/cis90/tinsam/1968.egg
```

```
/home/cis90/milhom/1968.egg
```

```
/home/cis90/bancar/1968.egg
```

*< snipped >*

```
/home/cis90/miljac/1968.egg
```

```
/home/cis90/stejad/basket/1968.egg
```

```
/home/cis90/simben/1968.egg
```

```
/home/cis90/hunbra/1968.egg
```

```
/home/cis90/specod/1968.egg
```

```
/home/cis90/temtyl/1968.egg
```

```
/home/cis90/watshe/1968.egg
```

```
/home/cis90/hawjus/1968.egg
```

```
/home/cis90/simreb/basket/1968.egg
```

```
/home/cis90/seasky/1968.egg
```

```
/home/cis90/brinic/1968.egg
```

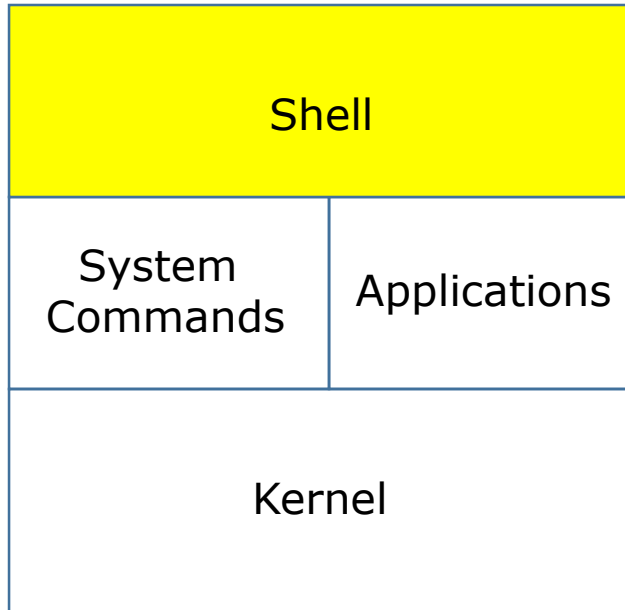
```
/home/cis90/simben $
```

*Note: Only the 1968.egg files were found!*

*That's because Benji has a file named 1968.egg in his home directory and the shell replaced \*egg with 1968.egg.*



# Prompt Step



**1) Prompt**

2) Parse

3) Search

4) Execute

5) Nap

6) Repeat







# Prompt Step (uses PS1 variable)

/home/cis90/simben \$

*bash using your PS1 variable creates and outputs your prompt which is written to your terminal device*

- Benji is using the bash shell. There are many other shells such as sh, ksh and csh. In `/etc/passwd` the last field in the line for his account determines the shell that is run when logging in.

```
/home/cis90/simben $ grep $LOGNAME /etc/passwd
simben90:x:1001:190:Benji Simms:/home/cis90/simben:/bin/bash
```

- The bash program resides in the `/bin` directory.

```
/home/cis90/simben $ ls -l /bin/bash
-rwxr-xr-x. 1 root root 874248 May 10 2012 /bin/bash
```

- The command prompt appearance is defined by the PS1 variable. You can output a prompt yourself using **echo \$PS1**

```
/home/cis90/simben $ echo $PS1
$PWD $
/home/cis90/simben $ echo $PWD $
/home/cis90/simben $
/home/cis90/simben $
```



# Prompt Step

*Note there is an invisible  
<newline> metacharacter at  
the end of the command*

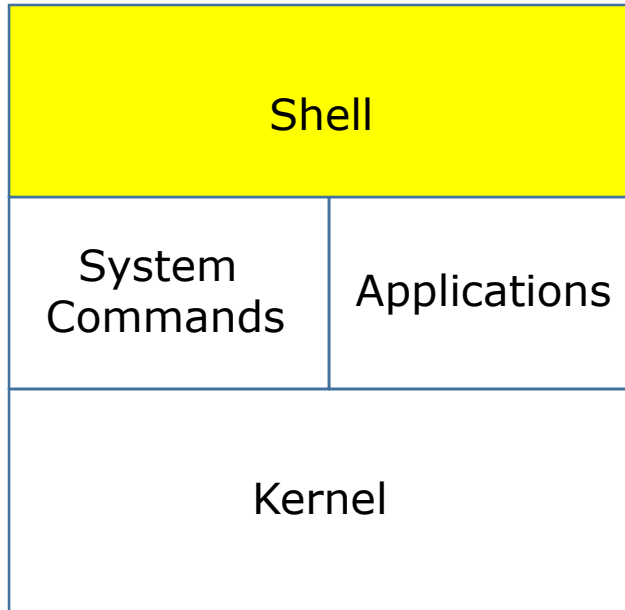
```
/home/cis90/simben $ find / -name *egg 2> /dev/null
```

*Benji types this find command in  
response to the shell prompt*

*The prompt step is not complete until the user type the Enter/Return key*



# Parse Step



- 1) Prompt
- 2) Parse**
- 3) Search
- 4) Execute
- 5) Nap
- 6) Repeat





# Parse Step

The shell uses spaces to separate options, arguments and redirection

**find** **/** **-name** **\*egg** **2>** **/dev/null**

The shell must expand filename expansion characters and variables during the parse step. Note there is an invisible <newline> metacharacter at the end of the command

## Parsing RESULTS:

Command: **find**

Options and arguments:

**/**

**-name**

**1968.egg**

This will be passed to the command (if the command can be located on the path)

Redirection:

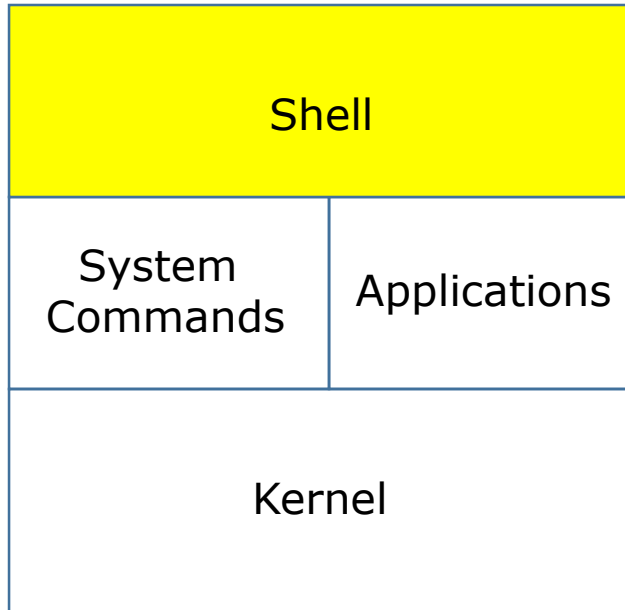
Connect **stderr** to **/dev/null** (the "bit bucket")

This will be handled by the shell. The command, if loaded, will not see this

Note: Because Benji had a 1968.egg file in his home directory, the shell expands \*egg to 1968.egg



# Search Step



1) Prompt

2) Parse

**3) Search**

4) Execute

5) Nap

6) Repeat







# Search Step (uses **PATH** variable)

Command: **find**

*The shell now must search, in order, every directory on Benji's path to locate the first occurrence of the **find** command.*

*Benji's path is defined by the value of his **PATH** variable*

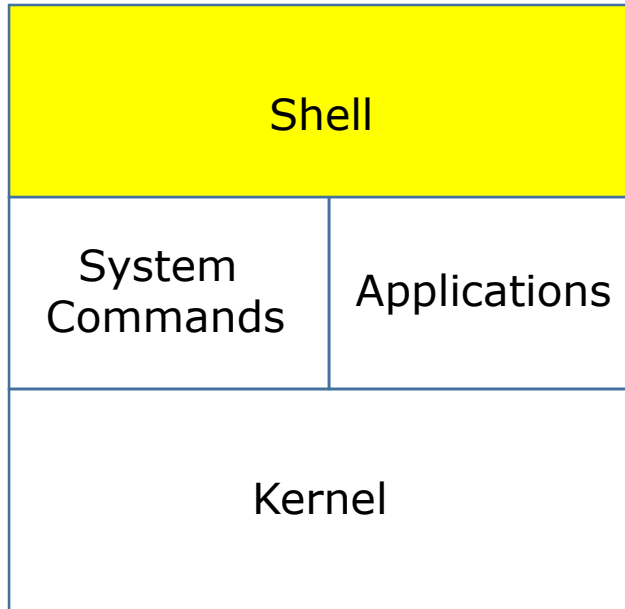
1<sup>st</sup> directory searched: /usr/local/bin  
2<sup>nd</sup> directory searched: /usr/bin  
3<sup>rd</sup> directory searched: /usr/local/sbin  
4<sup>th</sup> directory searched: /usr/sbin  
5<sup>th</sup> directory searched: /home/cis90/simben/../../bin  
6<sup>th</sup> directory searched: /home/cis90/simben/bin  
7<sup>th</sup> directory searched: .

*The shell  
locates the  
find command  
in the /usr/bin  
directory*

```
/home/cis90/simben $ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/cis90/simben/../../bin:/home/cis90/simben/bin:..
/home/cis90/simben $ type find
find is /usr/bin/find
```



# Execute Step

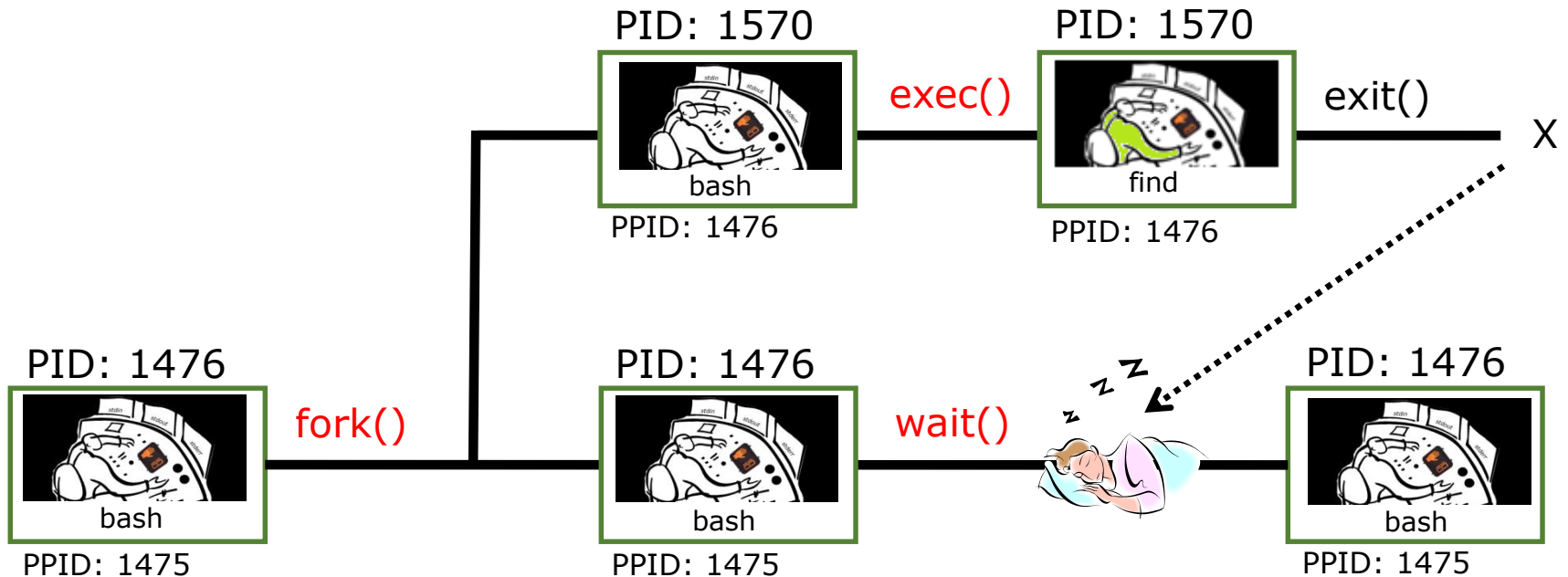


- 1) Prompt
- 2) Parse
- 3) Search
- 4) Execute**
- 5) Nap
- 6) Repeat





# Execute Step



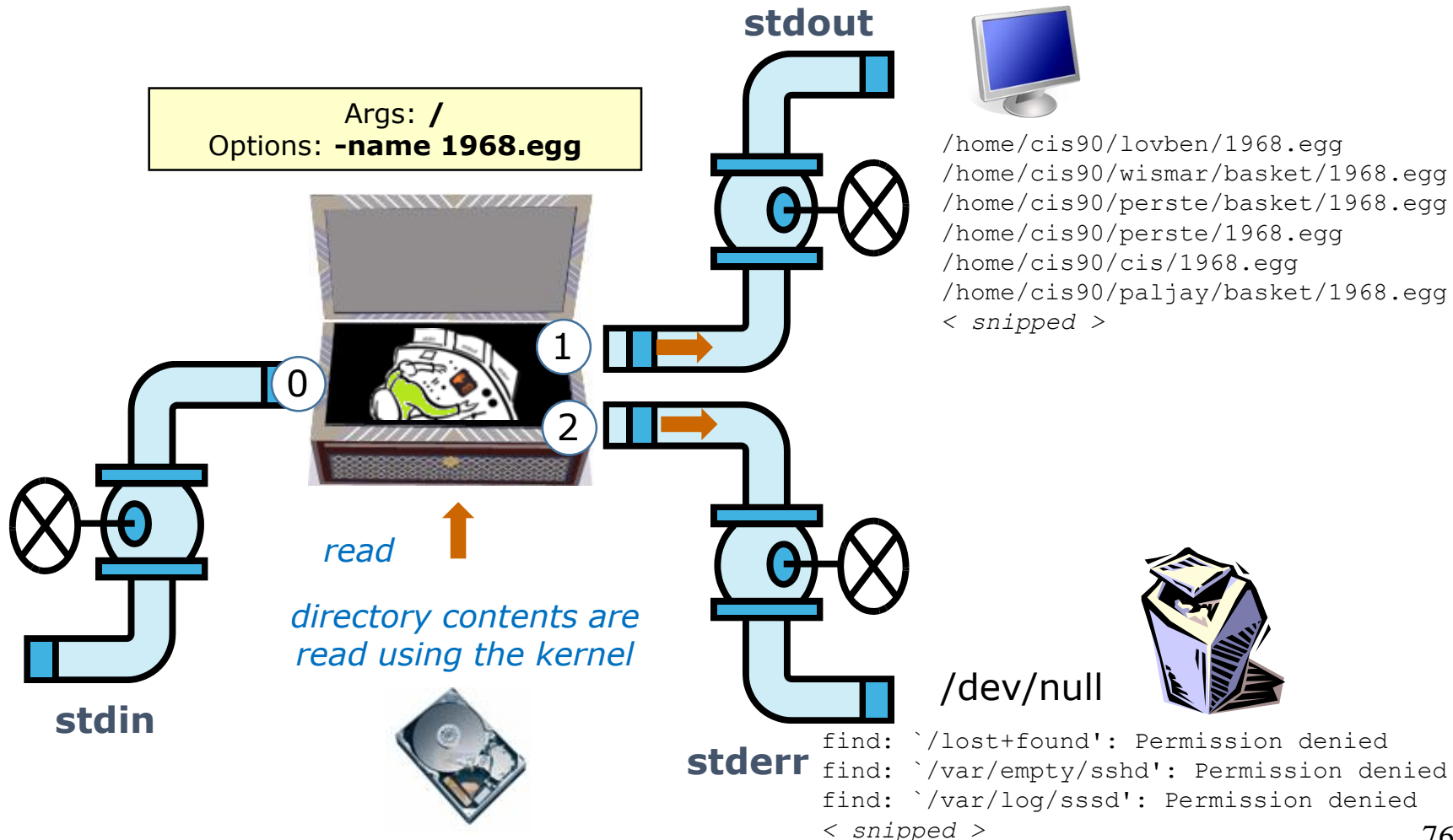
**bash** executes the **find** command by:

- 1) Cloning itself with a **fork()** system call to create a new child process.
- 2) With an **exec()** system call, the new child process is overlaid with the **find** code instructions.
- 3) **bash** sleeps by making a **wait()** system call while the **find** child process runs.
- 4) The child process makes an **exit()** system call when it has finished.
- 5) After that, the parent **bash** process wakes up and the child process is killed.

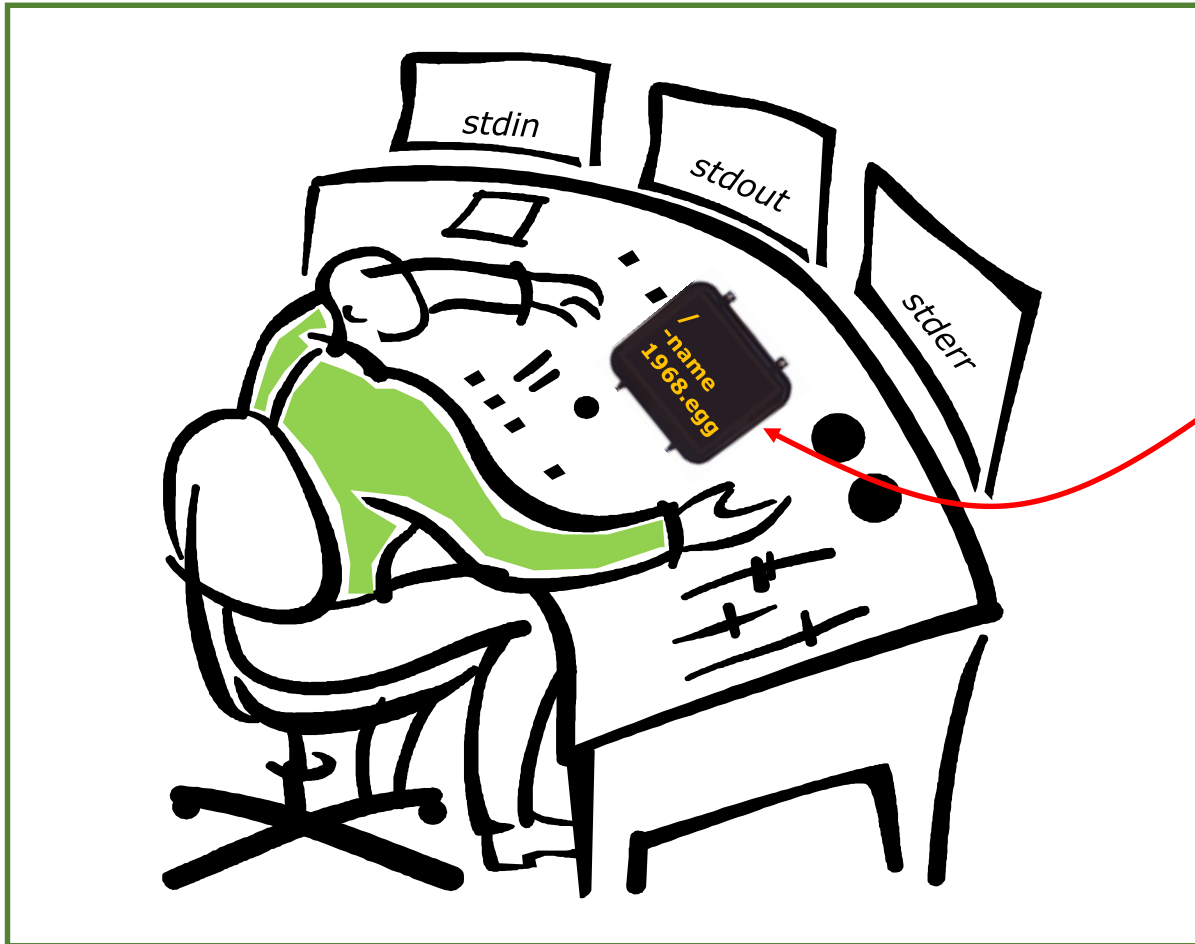


# Execute Step

```
/home/cis90/simben $ find / -name *egg 2> /dev/null
```



This is what the find process might look like



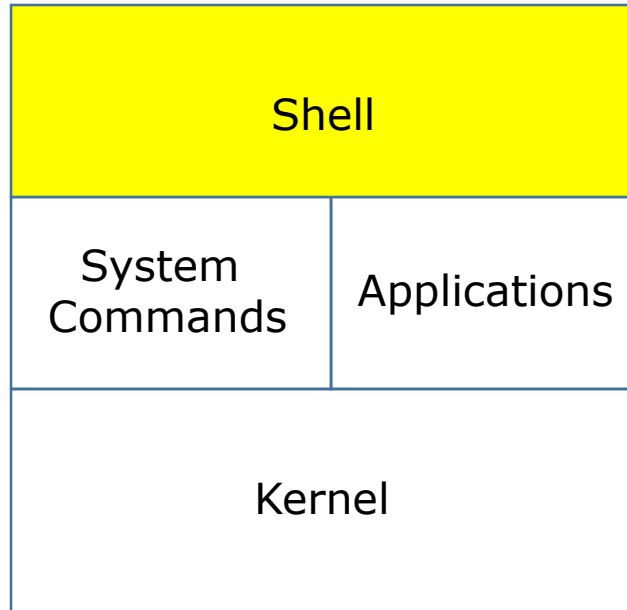
**A process:**

- Is provided with parsed & expanded options and arguments from the shell
- may read from **stdin**
- may write to **stdout**
- may write error messages to **stderr**
- and may get interrupted from time to time by a **signal**

*The **find** process is running*



# Nap Step



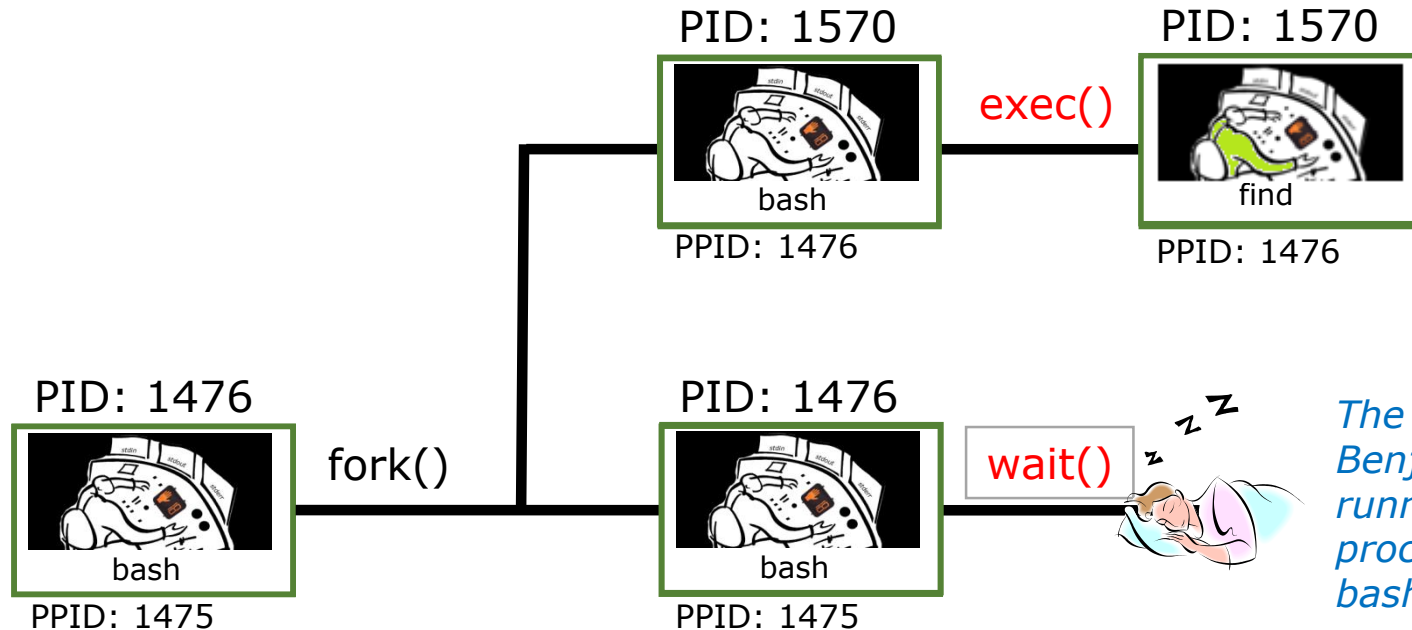
- 1) Prompt
- 2) Parse
- 3) Search
- 4) Execute
- 5) Nap**
- 6) Repeat







# Nap Step



The PS command shows Benji's **find** command is running as a child process while the parent bash shell sleeps

Sleeping

```
[rsimms@oslab ~]$ ps -l -u simben90
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
5	S	1001	1475	1470	0	80	0	-	3392	?	?	00:00:00	sshd
0	S	1001	1476	1475	0	80	0	-	1308	?	pts/1	00:00:00	bash
0	R	1001	1570	1476	40	80	0	-	1179	?	pts/1	00:00:00	find

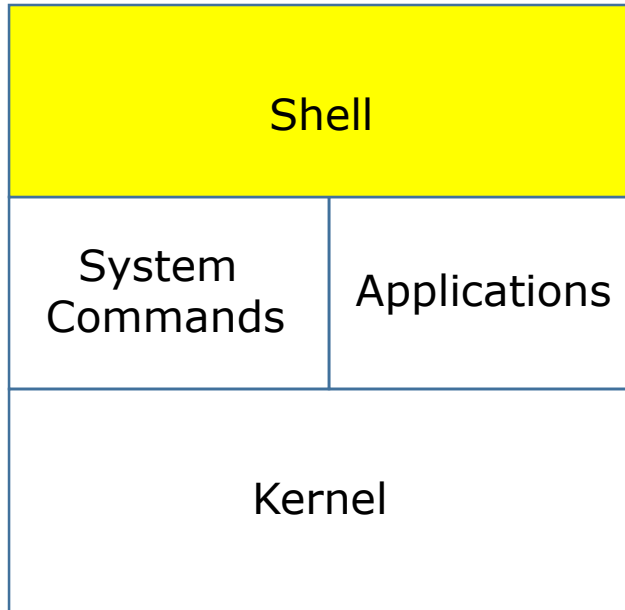
Parent

Child

Running



# Repeat Step

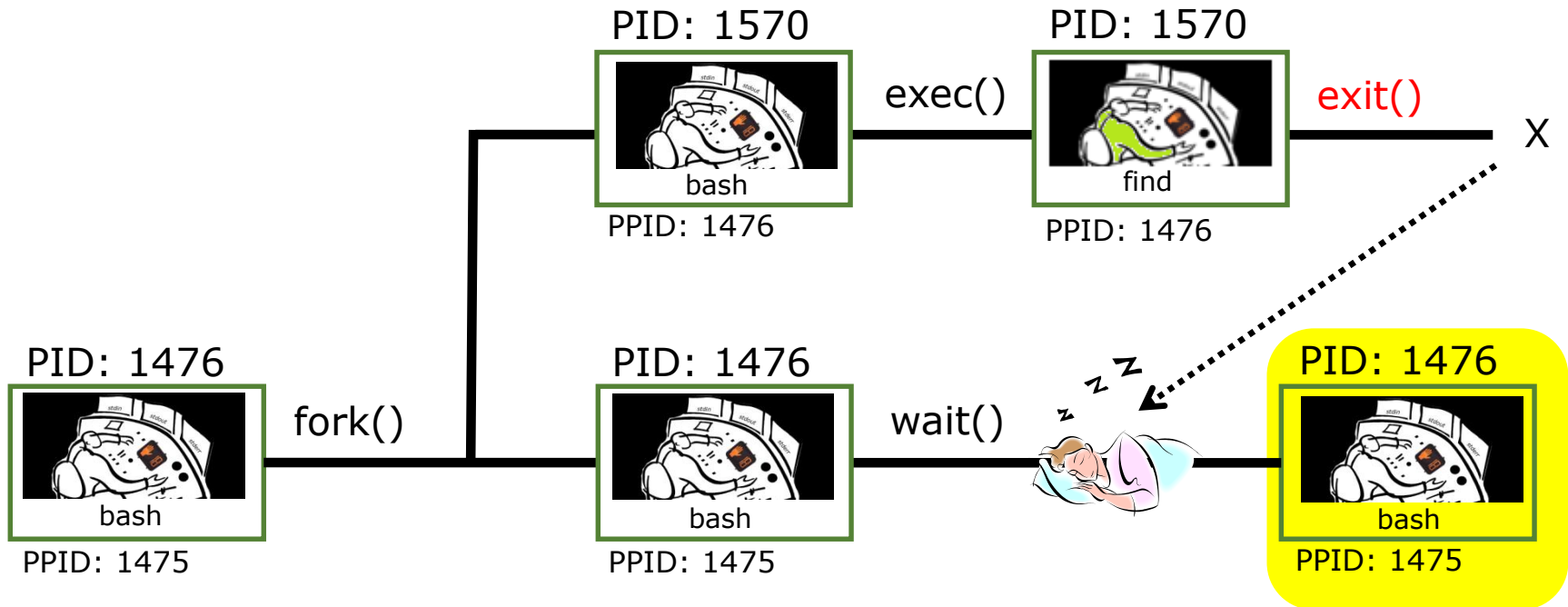


- 1) Prompt
- 2) Parse
- 3) Search
- 4) Execute
- 5) Nap
- 6) Repeat**





# Repeat Step



*The child process makes an **exit()** system call when it has finished. The parent bash process wakes up, the child process is killed and we are ready to start the process all over again with the next command.*

# Process activity

- Start a second login session and see if you can illustrate the parent sleeping while a child runs.
- In one session run: `grep -r "playing hot potato" /usr`
- In the second session use repeatedly: `ps -lu $LOGNAME`
- The **ps** output should show "parent" bash S=Sleeping while the "child" **grep** command is either R=Running or in D=Uninterruptible sleep (IO)

```
simben90@opus-iii:~$ grep -r "playing hot potato" /usr
grep: /usr/bin/staprun: Permission denied
grep: /usr/bin/chfn: Permission denied
grep: /usr/bin/chsh: Permission denied
grep: /usr/bin/ssh-agent: Permission denied
grep: /usr/bin/sudo: Permission denied
grep: /usr/bin/sudoreplay: Permission denied
grep: /usr/sbin/build-locale-archive: Permission denied
grep: /usr/sbin/glibc_post_upgrade.x86_64: Permission denied
grep: /usr/sbin/unix_update: Permission denied
grep: /usr/sbin/groupadd: Permission denied
grep: /usr/sbin/groupdel: Permission denied
grep: /usr/sbin/groupmems: Permission denied
```

```
simben90@opus-iii:~$ ps -lu $LOGNAME
```

	F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	1201	3163	3157	0	80	0	-	28881	do_wai	pts/2	00:00:00	bash	
4	S	1201	3202	3194	0	80	0	-	28881	do_wai	pts/1	00:00:00	bash	
0	R	1201	3252	3163	99	80	0	-	29687	-	pts/2	00:00:03	grep	
0	R	1201	3284	3202	0	80	0	-	37766	-	pts/1	00:00:00	ps	

```
/home/cis90/simben $ ps -lu $LOGNAME
```

	F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
4	S	1201	3163	3157	0	80	0	-	28881	do_wai	pts/2	00:00:00	bash	
4	S	1201	3202	3194	0	80	0	-	28881	do_wai	pts/1	00:00:00	bash	
0	R	1201	3252	3163	94	80	0	-	29687	-	pts/2	00:00:05	grep	
0	R	1201	3288	3202	0	80	0	-	37766	-	pts/1	00:00:00	ps	

```
/home/cis90/simben $
```

*Write your parent  
bash status and PID  
into the chat window*

**bash, pid=3163, status=S (sleeping)**



# ps command output (practice)

# Activity

```
rsimms@opus-ii:~/play/family-pid-tree
[rsimms@opus-ii family-pid-tree]$ ps -lu rsimms | grep -v sleep
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	2334	7595	0	80	0	-	28294	do_wai	pts/1	00:00:00	dora
0	S	1000	2335	2334	0	80	0	-	28294	do_wai	pts/1	00:00:00	ann
0	S	1000	2336	2334	0	80	0	-	28294	do_wai	pts/1	00:00:00	robert
0	S	1000	2339	2336	0	80	0	-	28294	do_wai	pts/1	00:00:00	jean
0	S	1000	2340	2336	0	80	0	-	28294	do_wai	pts/1	00:00:00	devi
0	S	1000	2343	2340	0	80	0	-	28294	do_wai	pts/1	00:00:00	yvonne
0	S	1000	2344	2340	0	80	0	-	28294	do_wai	pts/1	00:00:00	maria
0	S	1000	2347	2344	0	80	0	-	28294	do_wai	pts/1	00:00:00	michael
0	S	1000	2348	2344	0	80	0	-	28294	do_wai	pts/1	00:00:00	paul
0	R	1000	2390	7595	0	80	0	-	38840	-	pts/1	00:00:00	ps
0	S	1000	2391	7595	0	80	0	-	28177	pipe_w	pts/1	00:00:00	grep
5	S	1000	7594	7552	0	80	0	-	39759	poll_s	?	00:00:01	sshd
0	S	1000	7595	7594	0	80	0	-	28893	do_wai	pts/1	00:00:00	bash
1	S	1000	8518	1	0	80	0	-	30363	poll_s	?	00:12:29	gpg-agent

```
[rsimms@opus-ii family-pid-tree]$
```

Is the paul process running, stopped, sleeping or a zombie?  
*Put the process name and PID in the chat window*



# Activity

```
rsimms@opus-ii:~/play/family-pid-tree$ ps -lu rsimms | grep -v sleep
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	2334	7595	0	80	0	-	28294	do_wai	pts/1	00:00:00	dora
0	S	1000	2335	2334	0	80	0	-	28294	do_wai	pts/1	00:00:00	ann
0	S	1000	2336	2334	0	80	0	-	28294	do_wai	pts/1	00:00:00	robert
0	S	1000	2339	2336	0	80	0	-	28294	do_wai	pts/1	00:00:00	jean
0	S	1000	2340	2336	0	80	0	-	28294	do_wai	pts/1	00:00:00	devi
0	S	1000	2343	2340	0	80	0	-	28294	do_wai	pts/1	00:00:00	yvonne
0	S	1000	2344	2340	0	80	0	-	28294	do_wai	pts/1	00:00:00	maria
0	S	1000	2347	2344	0	80	0	-	28294	do_wai	pts/1	00:00:00	michael
0	S	1000	2348	2344	0	80	0	-	28294	do_wai	pts/1	00:00:00	paul
0	R	1000	2390	7595	0	80	0	-	38840	-	pts/1	00:00:00	ps
0	S	1000	2391	7595	0	80	0	-	28177	pipe_w	pts/1	00:00:00	grep
5	S	1000	7594	7552	0	80	0	-	39759	poll_s	?	00:00:01	sshd
0	S	1000	7595	7594	0	80	0	-	28893	do_wai	pts/1	00:00:00	bash
1	S	1000	8518	1	0	80	0	-	30363	poll_s	?	00:12:29	gpg-agent

```
[rsimms@opus-ii family-pid-tree]$
```

Which process is the "great-grandparent" of the paul process?  
*Put the process name and PID in the chat window*

# Activity

```
rsimms@opus-ii:~/play/family-pid-tree
[rsimms@opus-ii family-pid-tree]$ ps -lu rsimms | grep -v sleep
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	2334	7595	0	80	0	-	28294	do_wai	pts/1	00:00:00	dora
0	S	1000	2335	2334	0	80	0	-	28294	do_wai	pts/1	00:00:00	ann
0	S	1000	2336	2334	0	80	0	-	28294	do_wai	pts/1	00:00:00	robert
0	S	1000	2339	2336	0	80	0	-	28294	do_wai	pts/1	00:00:00	jean
0	S	1000	2340	2336	0	80	0	-	28294	do_wai	pts/1	00:00:00	devi
0	S	1000	2343	2340	0	80	0	-	28294	do_wai	pts/1	00:00:00	yvonne
0	S	1000	2344	2340	0	80	0	-	28294	do_wai	pts/1	00:00:00	maria
0	S	1000	2347	2344	0	80	0	-	28294	do_wai	pts/1	00:00:00	michael
0	S	1000	2348	2344	0	80	0	-	28294	do_wai	pts/1	00:00:00	paul
0	R	1000	2390	7595	0	80	0	-	38840	-	pts/1	00:00:00	ps
0	S	1000	2391	7595	0	80	0	-	28177	pipe_w	pts/1	00:00:00	grep
5	S	1000	7594	7552	0	80	0	-	39759	poll_s	?	00:00:01	sshd
0	S	1000	7595	7594	0	80	0	-	28893	do_wai	pts/1	00:00:00	bash
1	S	1000	8518	1	0	80	0	-	30363	poll_s	?	00:12:29	gpg-agent

```
[rsimms@opus-ii family-pid-tree]$
```

Which processes are children of the devi process?  
*Put the child process names and PIDs in the chat window*

# Activity

```
rsimms@opus-ii:~/play/family-pid-tree$ ps -lu rsimms | grep -v sleep
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	2334	7595	0	80	0	-	28294	do_wai	pts/1	00:00:00	dora
0	S	1000	2335	2334	0	80	0	-	28294	do_wai	pts/1	00:00:00	ann
0	S	1000	2336	2334	0	80	0	-	28294	do_wai	pts/1	00:00:00	robert
0	S	1000	2339	2336	0	80	0	-	28294	do_wai	pts/1	00:00:00	jean
0	S	1000	2340	2336	0	80	0	-	28294	do_wai	pts/1	00:00:00	devi
0	S	1000	2343	2340	0	80	0	-	28294	do_wai	pts/1	00:00:00	yvonne
0	S	1000	2344	2340	0	80	0	-	28294	do_wai	pts/1	00:00:00	maria
0	S	1000	2347	2344	0	80	0	-	28294	do_wai	pts/1	00:00:00	michael
0	S	1000	2348	2344	0	80	0	-	28294	do_wai	pts/1	00:00:00	paul
0	R	1000	2390	7595	0	80	0	-	38840	-	pts/1	00:00:00	ps
0	S	1000	2391	7595	0	80	0	-	28177	pipe_w	pts/1	00:00:00	grep
5	S	1000	7594	7552	0	80	0	-	39759	poll_s	?	00:00:01	sshd
0	S	1000	7595	7594	0	80	0	-	28893	do_wai	pts/1	00:00:00	bash
1	S	1000	8518	1	0	80	0	-	30363	poll_s	?	00:12:29	gpg-agent

```
[rsimms@opus-ii family-pid-tree]$
```

Does the jean process have a "sibling" process (same parent)?  
*Put the "sibling" process name and PID in the chat window*

```
rsimms@opus-ii:~/play/family-pid-tree
[rsimms@opus-ii family-pid-tree]$ ps -lu rsimms | grep -v sleep
F S  UID    PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000    2334  7595  0  80   0 - 28294 do_wai pts/1      00:00:00 dora
0 S  1000    2335  2334  0  80   0 - 28294 do_wai pts/1      00:00:00 ann
0 S  1000    2336  2334  0  80   0 - 28294 do_wai pts/1      00:00:00 robert
0 S  1000    2339  2336  0  80   0 - 28294 do_wai pts/1      00:00:00 jean
0 S  1000    2340  2336  0  80   0 - 28294 do_wai pts/1      00:00:00 devi
0 S  1000    2343  2340  0  80   0 - 28294 do_wai pts/1      00:00:00 yvonne
0 S  1000    2344  2340  0  80   0 - 28294 do_wai pts/1      00:00:00 maria
0 S  1000    2347  2344  0  80   0 - 28294 do_wai pts/1      00:00:00 michael
0 S  1000    2348  2344  0  80   0 - 28294 do_wai pts/1      00:00:00 paul
0 R  1000    2390  7595  0  80   0 - 38840 -      pts/1      00:00:00 ps
0 S  1000    2391  7595  0  80   0 - 28177 pipe_w pts/1      00:00:00 grep
5 S  1000    7594  7552  0  80   0 - 39759 poll_s ?          00:00:01 sshd
0 S  1000    7595  7594  0  80   0 - 28893 do_wai pts/1      00:00:00 bash
1 S  1000    8518    1  0  80   0 - 30363 poll_s ?          00:12:29 gpg-agent
[rsimms@opus-ii family-pid-tree]$
```

Is the paul process running, stopped, sleeping or a zombie?

*Put the process name and PID in the chat window*  
sleeping

Which process is the "great-grandparent" of the paul process?

*Put the process name and PID in the chat window*  
robert 2336

Which processes are children of the devi process?

*Put the child process names and PIDs in the chat window*  
yvonne 2343 and maria 2344

Does the jean process have a "sibling" process (same parent)?

*Put the "sibling" process name and PID in the chat window*  
devi 2340



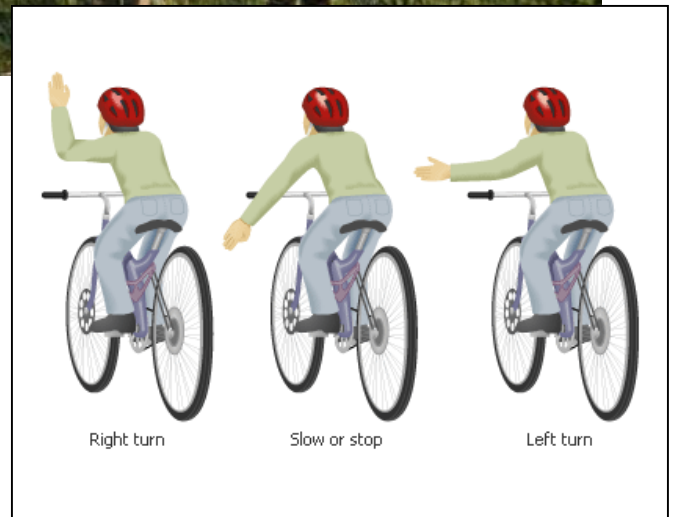
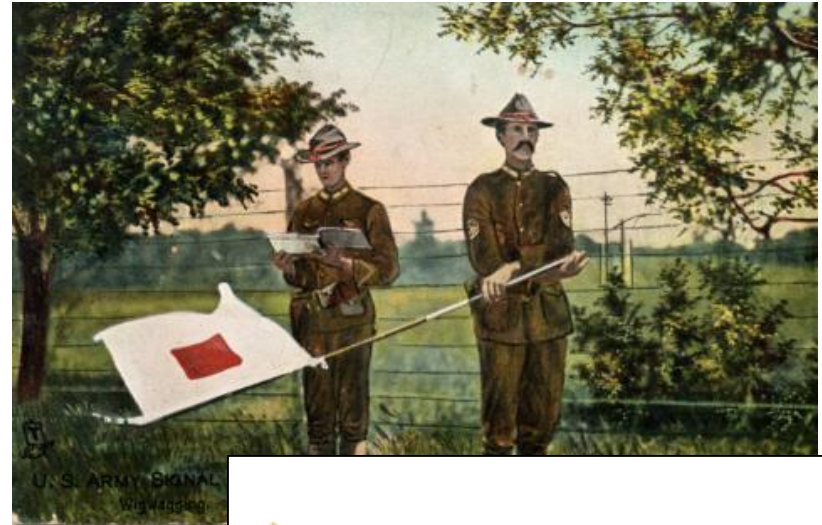
# Signals (Review)



# Signals

PLATE 4

COMMERCIAL CODE SIGNALS					
<p>EXAMPLES OF THE SEVERAL HOISTS WHICH CAN BE MADE HAVING TWO, THREE, OR FOUR FLAGS. When a word contains two letters of the same name, the second time of its occurrence it must begin or be in the 2nd Hoist; and on its 3rd occurrence, it must begin or be in the 3rd Hoist.</p>					
URGENT & IMPORTANT SIGNALS		COMPASS SIGNALS 3 FLAGS			
CODE FLAG OVER 1 FLAG OR 2 FLAG SIGNALS					
CODE FLAG P	A	A	Q	K	X
"I Am about to Sail"	"Do Not"		N 1/2 E	S 57° W	
LATITUDE & LONGITUDE SIGNALS		CODE FLAG OVER 2 FLAGS			
CODE FLAG A	Q	CODE FLAG E	Q	Y	Z
OR H	OR H	OR Y	OR Y	OR Y	OR Y
12° Latitude	North Latitude	23° Longitude	East Longitude		
NUMERAL TABLE		GENERAL VOCABULARY			
CODE FLAG UNDER 2 FLAGS		3 FLAG SIGNAL			
Y	S	I	X	K	
CODE FLAG 10,000		Tons of Coal			
ALPHABETICAL SPELLING TABLE		GEOGRAPHICAL SIGNALS ALPHABETICAL ORDER			
J	C	A	E	Y	Z
O	B	S			
H	D	F			
N	N	P			
John	Abb	off			
		NAMES OF VESSELS FROM CODE LIST			
		4 FLAG SIGNAL			
		H	C	L	B
		Grunder of Glasgow			
		1058 Tons N° 52636			





# Unix Signals

```
/home/cis90/rodduk $ kill -l      Use kill -l to see all signals
 1) SIGHUP          2) SIGINT          3) SIGQUIT          4) SIGILL
 5) SIGTRAP         6) SIGABRT         7) SIGBUS           8) SIGFPE
 9) SIGKILL        10) SIGUSR1        11) SIGSEGV         12) SIGUSR2
13) SIGPIPE        14) SIGALRM        15) SIGTERM         16) SIGSTKFLT
17) SIGCHLD        18) SIGCONT        19) SIGSTOP         20) SIGTSTP
21) SIGTTIN        22) SIGTTOU        23) SIGURG          24) SIGXCPU
25) SIGXFSZ        26) SIGVTALRM      27) SIGPROF         28) SIGWINCH
29) SIGIO          30) SIGPWR          31) SIGSYS          34) SIGRTMIN
35) SIGRTMIN+1     36) SIGRTMIN+2     37) SIGRTMIN+3     38) SIGRTMIN+4
39) SIGRTMIN+5     40) SIGRTMIN+6     41) SIGRTMIN+7     42) SIGRTMIN+8
43) SIGRTMIN+9     44) SIGRTMIN+10    45) SIGRTMIN+11    46) SIGRTMIN+12
47) SIGRTMIN+13    48) SIGRTMIN+14    49) SIGRTMIN+15    50) SIGRTMAX-14
51) SIGRTMAX-13    52) SIGRTMAX-12    53) SIGRTMAX-11    54) SIGRTMAX-10
55) SIGRTMAX-9     56) SIGRTMAX-8     57) SIGRTMAX-7     58) SIGRTMAX-6
59) SIGRTMAX-5     60) SIGRTMAX-4     61) SIGRTMAX-3     62) SIGRTMAX-2
63) SIGRTMAX-1     64) SIGRTMAX
/home/cis90/rodduk $
```

*Unix signals are integers in the range of 1-64*

# Unix Signals

SIGHUP	1	Hangup (POSIX)	
SIGINT	2	Terminal interrupt (ANSI)	Ctrl-C
SIGQUIT	3	Terminal quit (POSIX)	Ctrl-\
SIGILL	4	Illegal instruction (ANSI)	
SIGTRAP	5	Trace trap (POSIX)	
SIGIOT	6	IOT Trap (4.2 BSD)	
SIGBUS	7	BUS error (4.2 BSD)	
SIGFPE	8	Floating point exception (ANSI)	
SIGKILL	9	Kill (can't be caught or ignored) (POSIX)	
SIGUSR1	10	User defined signal 1 (POSIX)	
SIGSEGV	11	Invalid memory segment access (ANSI)	
SIGUSR2	12	User defined signal 2 (POSIX)	
SIGPIPE	13	Write on a pipe with no reader, Broken pipe (POSIX)	
SIGALRM	14	Alarm clock (POSIX)	
SIGTERM	15	Termination (ANSI)	

*Some signals can be sent using keystrokes, e.g. Ctrl-C or Ctrl-\*

## Unix Signals

SIGSTKFLT	16	Stack fault
SIGCHLD	17	Child process has stopped or exited, changed (POSIX)
SIGCONT	18	Continue executing, if stopped (POSIX)
SIGSTOP	19	Stop executing(can't be caught or ignored) (POSIX)
SIGTSTP	20	Terminal stop signal (POSIX) <b>Ctrl-Z or Ctrl-F</b>
SIGTTIN	21	Background process trying to read, from TTY (POSIX)
SIGTTOU	22	Background process trying to write, to TTY (POSIX)
SIGURG	23	Urgent condition on socket (4.2 BSD)
SIGXCPU	24	CPU limit exceeded (4.2 BSD)
SIGXFSZ	25	File size limit exceeded (4.2 BSD)
SIGVTALRM	26	Virtual alarm clock (4.2 BSD)
SIGPROF	27	Profiling alarm clock (4.2 BSD)
SIGWINCH	28	Window size change (4.3 BSD, Sun)
SIGIO	29	I/O now possible (4.2 BSD)
SIGPWR	30	Power failure restart (System V)

*Use kill -l to see all signals*

# Signals

*Signals are asynchronous messages sent to processes*

Asynchronous  
means it can  
happen at any  
time

*The messages  
are just integers  
in the range of  
1 to 64*



The executing process above is receiving the 3 (SIGQUIT) signal

# How to send a signal to a process

## Signals are asynchronous messages sent to processes

Signals are sent:

- Using the **kill** command: **kill -# <PID>**
  - Where # is the signal number (1-64).
  - <PID> is the process ID.
  - if no signal number is specified, SIGTERM 15 is sent.
- Or using the **killall** command: **killall -# <name of process>**
  - Where # is the signal number.
  - if no signal number is specified, SIGTERM 15 is sent.
- Or Using special **keystrokes** (e.g. Ctrl-C for SIGINT/2)
  - Limited to just a few signals.
  - Sent to the process running in the foreground.
  - Use **stty -a** to see keystroke assignments.

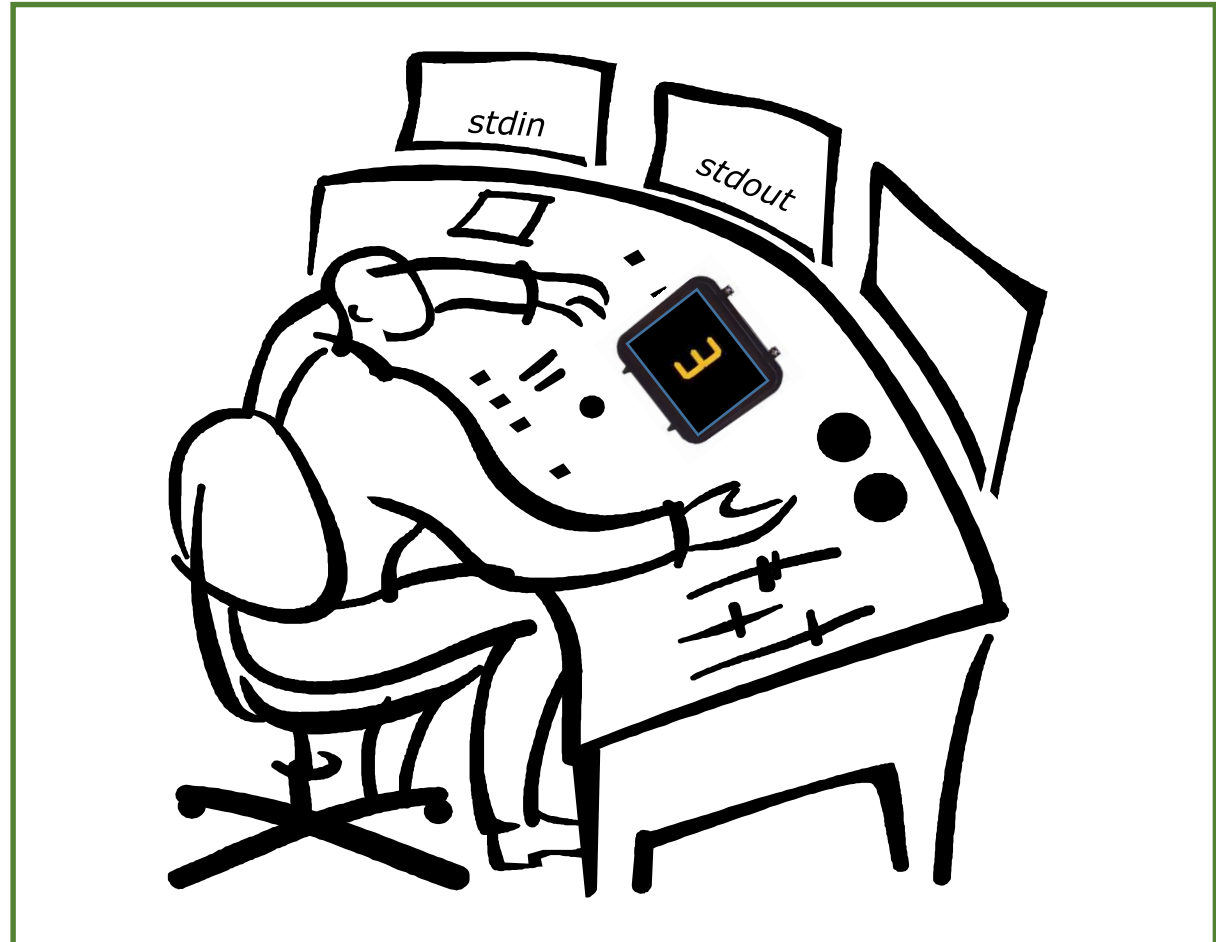
*Use kill -l to see all signals*

*If you are stuck on the final exam trying to figure out how to send a signal to a process you have come to the right place! :)*

# Signals

When a process receives a signal it will do one of the following:

- Ignore it.
- Take the default action (die).
- Execute some predefined function.

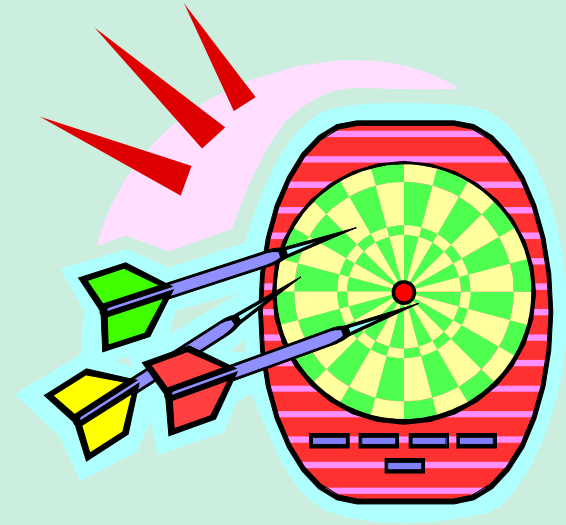


The executing process above is receiving the 3 (SIGQUIT) signal





# Target Practice



## Activity

- 1) Run the **annoy** program
- 2) Try sending it a SIGINT with **Ctrl-C**
- 3) Try sending it a SIGQUIT with **Ctrl-\**
- 4) Bring up another terminal and try signals 1 through 64
  - Use **ps -u \$LOGNAME** to find the **annoy** *PID*
  - Try **kill -1 <PID>**
  - Try **kill -2 <PID>**
  - Try **kill -3 <PID>**
  - *and so forth ...*
  - OR
  - Try **killall -1 annoy**
  - Try **killall -2 annoy**
  - Try **killall -3 annoy**
  - *and so forth ...*

*Write the signals that kill **annoy** into the chat window*

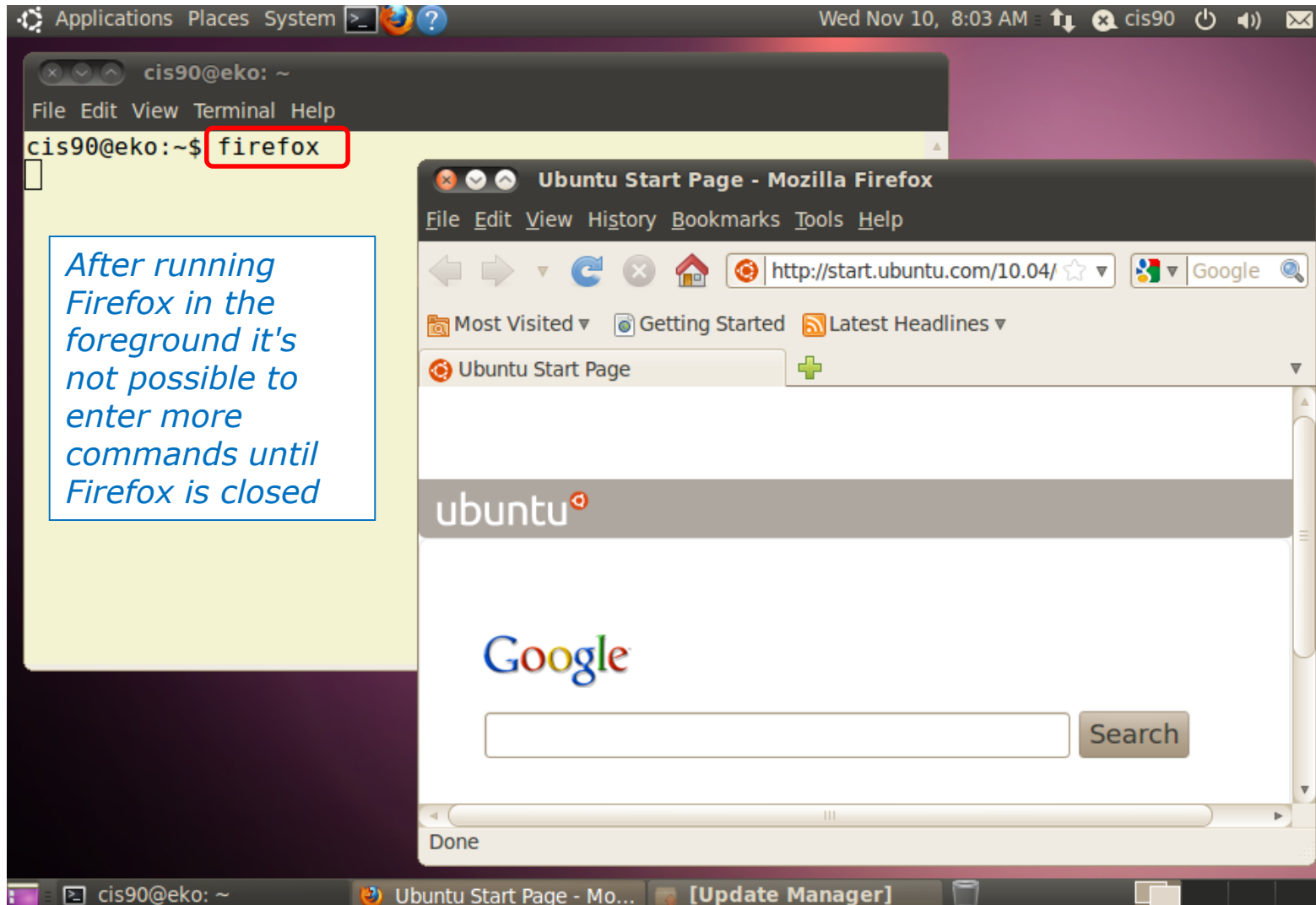


# Using &

to run a command  
in the background

## Job Control

Using **&** to run a command in the background



## Job Control

Using **&** to run a command in the background

The screenshot shows a Linux desktop environment. In the foreground, a terminal window titled 'cis90@eko: ~' displays the following commands and output:

```

cis90@eko:~$ firefox
cis90@eko:~$ firefox &
[1] 1465
cis90@eko:~$ ps
  PID TTY          TIME CMD
 1370 pts/0    00:00:00 bash
  1465 pts/0    00:00:00 firefox
  1470 pts/0    00:00:00 run-moz
  1474 pts/0    00:00:01 firefox
  1489 pts/0    00:00:00 ps
cis90@eko:~$
  
```

The command 'firefox &' is highlighted with a red box. Below the terminal output, a blue-bordered box contains the text: "After running Firefox in the background, it is still possible to enter more commands."

In the background, a Mozilla Firefox browser window titled 'Ubuntu Start Page - Mozilla Firefox' is open, displaying the Ubuntu Start Page at 'http://start.ubuntu.com/1'. The browser's address bar, search bar, and navigation buttons are visible.

**&** append to a command to run it in the background

### Example 1

```
/home/cis90/simben $ grep -r "playing hot potato" /usr 2> /dev/null
```

 No prompt, bash is asleep.

*For long running commands or scripts you must wait for the command to finish before you type more commands*

### Example 2

```
/home/cis90/simben $ grep -r "playing hot potato" /usr /opt 2> /dev/null &
[1] 7921
/home/cis90/simben $ date
Fri Apr 13 13:44:00 PDT 2018
```



*Hit enter to get the prompt and continue working while the find command runs in the background*





# Job Control (Review)

# Job Control

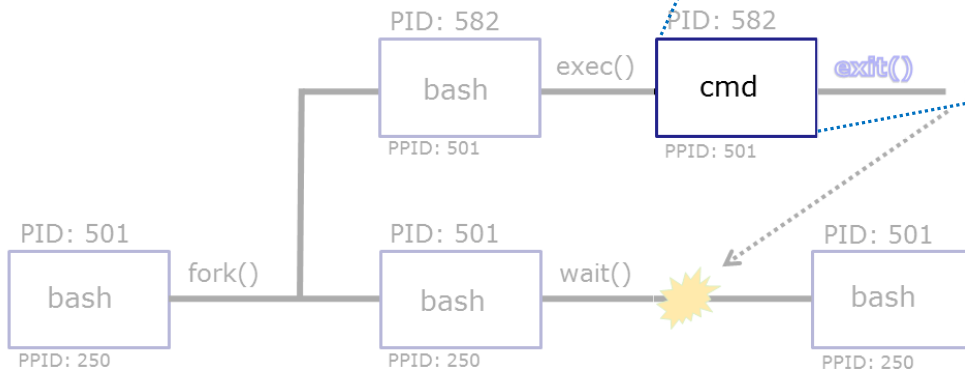
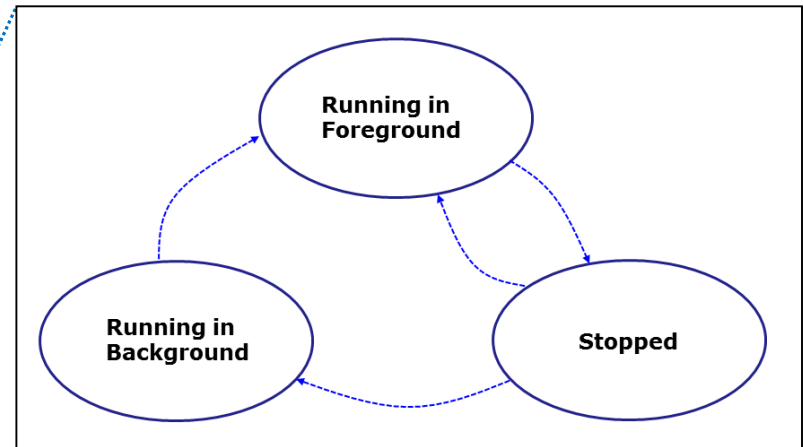
## A feature of the bash shell

<b>&amp;</b>	Append to a command to run it in the background
<b>bg</b>	Resumes a suspended job in the background
<b>fg</b>	Brings the most recent background process to the foreground
<b>jobs</b>	Lists all background jobs

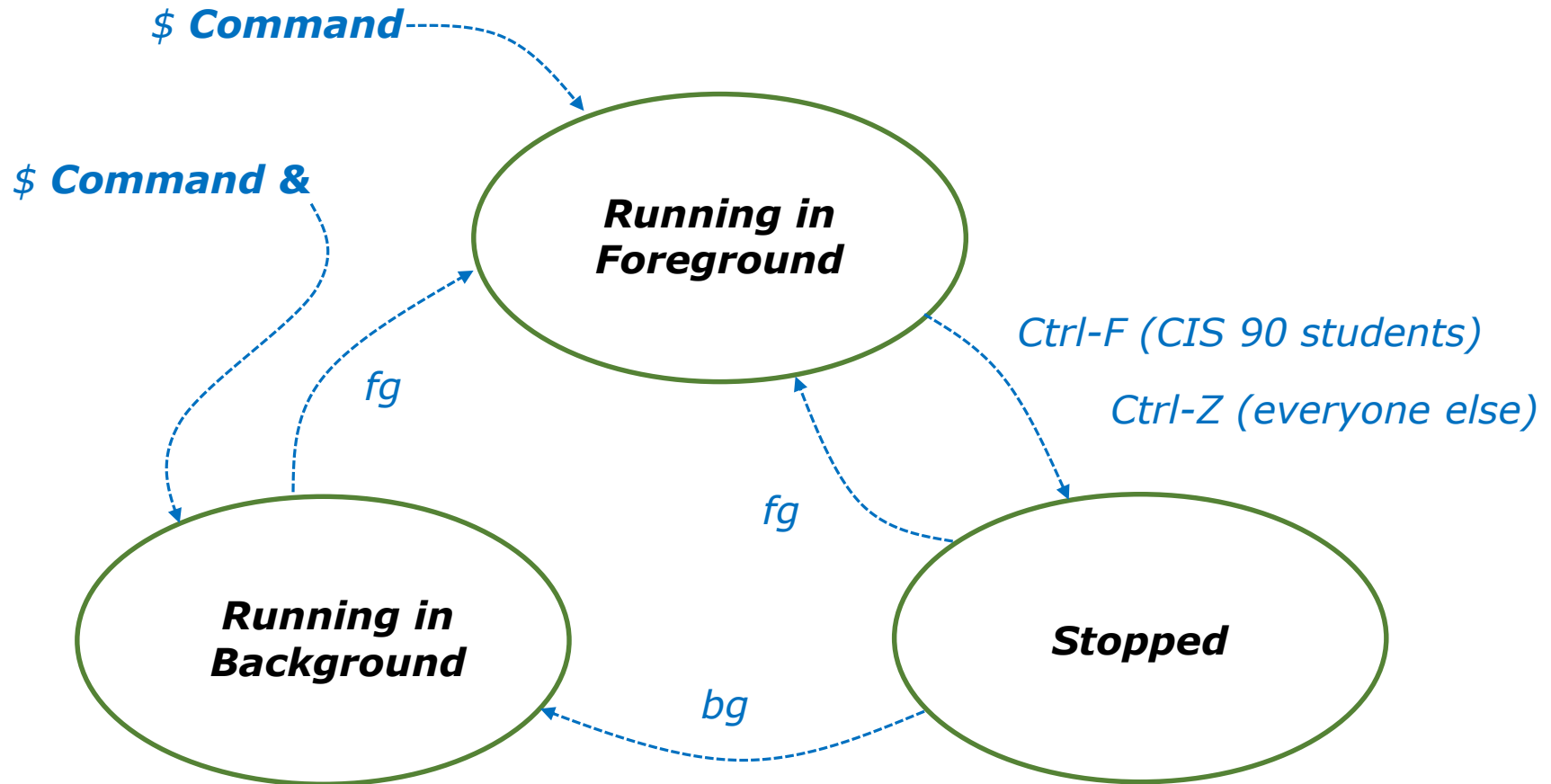
*Use **jobs**, **bg**, **fg** to list and resume jobs in the foreground or background*

## Job Control A feature of the bash shell

When a process is **running** (status=R) the user can **stop** it (status=T) and choose whether it runs in the **background** or **foreground**



## Job Control A feature of the bash shell



Use the **jobs** command to view  
stopped and background jobs

# Job Control

## Find out with keystroke combination is configured to suspend a process

```
/home/cis90ol/simmsben $ stty -a
speed 38400 baud; rows 24; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swch = <undef>; start = ^Q; stop = ^S; susp = ^F; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstopb cread -clocal -crtscts -cdtrdsr
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel -iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echoprt
echoctl echoke
/home/cis90ol/simmsben $
```

*In this case it is Ctrl-F that will  
be used to suspend a process*

*Put how your suspend keystrokes are  
configured in the chat window*

## Job Control

### Managing jobs

```
/home/cis90ol/simmsben $ sleep 120
Ctrl-Z or Ctrl-F (to suspend process)
[1]+  Stopped                  sleep 120
```

```
/home/cis90ol/simmsben $ sleep 110
Ctrl-Z or Ctrl-F (to suspend process)
[2]+  Stopped                  sleep 110
```

```
/home/cis90ol/simmsben $ sleep 100
Ctrl-Z or Ctrl-F (to suspend process)
[3]+  Stopped                  sleep 100
```

```
/home/cis90ol/simmsben $ jobs
[1]  Stopped                  sleep 120
[2]-  Stopped                  sleep 110
[3]+  Stopped                  sleep 100
```

*Lets start up 3 sleep commands and suspend each of them.*

*Note: The sleep command is a simple way to run a command that will take awhile to finish.*

***sleep 120** will last 120 seconds before it is finished.*



## Job Control

### Managing jobs

```
/home/cis90ol/simmsben $ jobs
```

```
[1]      Stopped                sleep 120
[2]-     Stopped                sleep 110
[3]+     Stopped                sleep 100
```

```
/home/cis90ol/simmsben $ ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1082	5364	5363	0	75	0	-	1168	wait	pts/2	00:00:00	bash
0	T	1082	5452	5364	0	75	0	-	929	finish	pts/2	00:00:00	sleep
0	T	1082	5453	5364	0	75	0	-	929	finish	pts/2	00:00:00	sleep
0	T	1082	5454	5364	0	75	0	-	929	finish	pts/2	00:00:00	sleep
0	R	1082	5459	5364	0	77	0	-	1054	-	pts/2	00:00:00	ps

*Note, all three processes are sTopped*

## Job Control

### Managing jobs

```
/home/cis90ol/simmsben $ bg 2 Let's resume job 2 in the background
```

```
[2]- sleep 110 &
```

```
/home/cis90ol/simmsben $ jobs
```

```
[1]- Stopped sleep 120
```

```
[2] Running sleep 110 &
```

```
[3]+ Stopped sleep 100
```

```
/home/cis90ol/simmsben $ bg 1 Let's resume job 1 in the background
```

```
[1]- sleep 120 &
```

```
/home/cis90ol/simmsben $ jobs
```

```
[1] Running sleep 120 &
```

```
[2]- Running sleep 110 &
```

```
[3]+ Stopped sleep 100
```

```
/home/cis90ol/simmsben $ fg 3 Let's resume job 1 in the foreground
```

```
sleep 100
```

*At this point we lose control of the keyboard again  
until sleep 100 is finished*

## Job Control

### Managing jobs

```
/home/cis90ol/simmsben $ jobs  
[1]-  Done  
sleep 120  
[2]+  Done  
sleep 110
```

*Background jobs are  
all done!*



# Load Balancing & Scheduling (Review)

## Load Balancing with **at** command

So that the multiprocessing CPU on a UNIX system does not get overloaded, some processes need to be run during low peak hours such as early in the morning or later in the day.

The **at** command reads from **stdin** for a list of commands to run, and begins running them at the time of day specified as the first argument.

Any output sent to **stdout** or **stderr** by the list of commands will be emailed to the user unless redirected elsewhere.



## Tools for your toolbox

NEW

**at** - schedule a job to run in the future

NEW

**at -c <jobnum>** - view a scheduled job

NEW

**atq** - list queue of pending jobs

NEW

**atrm** - remove a pending job



# at command

## Basic syntax

(see man page for the rest of the story)

**at <time>**     *Note: at reads commands to execute from stdin*

Example 1:

**at 3PM wed**

at> **cp -v letter letter.bak**

at> <EOT>

job 2482 at Wed Apr 17 15:00:00 2019

*Use Ctrl-D (end of file)  
to end.*

Example 2:

**echo "cp -v letter letter.bak" | at 6:51 am**

job 2483 at Wed Apr 10 06:51:00 2019

Example 3:

**echo "cp -v letter letter.bak" > commands**

**at < commands 11:59 pm**

job 2484 at Wed Apr 10 23:59:00 2019

## at command

Specifying future time examples:

`at now + 5 minutes`

`at now + 2 hours`

`at now + 1 week`

`at 1:00AM`

`at 3:00PM wednesday`

`at 12:00AM 12/25/2019`

`at noon`

`at midnight`

`at teatime`

## at examples

```
at 12:00 am thursday  
  chmod 700 /home/rsimms/turnin
```

*Lock and unlock  
a directory*

```
at 9:00 am thursday  
  chmod 750 /home/rsimms/turnin
```

*Turn in  
a lab*

```
at 11:59pm  
  cat files.out bigshell > lab08  
  cp lab08 /home/rsimms/turnin/cis90/lab08.$LOGNAME
```

```
at 2:50pm tuesday  
  cp /etc/nologin.bak /etc/nologin  
  shutdown -P +10
```

*Shutdown a  
system*

## at job management

```
/home/cis90/simben $ echo chmod 000 letter | at 3:00pm
job 878 at 2014-11-03 15:00
/home/cis90/simben $ echo chmod 644 letter | at 3:05pm
job 879 at 2014-11-03 15:05
/home/cis90/simben $ echo chmod 640 letter | at 1:00am friday
job 880 at 2014-11-07 01:00
```

```
/home/cis90/simben $ atq
879      2014-11-03 15:05 a simben90
880      2014-11-07 01:00 a simben90
878      2014-11-03 15:00 a simben90
```

*The **atq** command lists the queue of pending jobs scheduled to run in the future.*

```
/home/cis90/simben $ atrm 879
/home/cis90/simben $ atq
880      2014-11-07 01:00 a simben90
878      2014-11-03 15:00 a simben90
```

*The **atrm** command is used to remove jobs from the queue.*

```
/home/cis90/simben $ atrm 878 880
/home/cis90/simben $ atq
/home/cis90/simben $
```

## at command output handling

```
/home/cis90/simben $ at now + 1 minute
at> kitty letter
at> <EOT>
job 150 at 2011-04-20 10:47
```

*Oops, specified a non-existent command to run in the future (**kitty** should have been **cat**)*

```
/home/cis90/simben $ atq
150      2011-04-20 10:47 a simmsben
/home/cis90ol/simmsben $ atq
```

```
/home/cis90/simben $ mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/simben": 1 message 1 new
>N 1 simben@Opus.cabrillo.edu Wed Apr 20 10:47 16/709 "Output from your job "
& 1
Message 1:
From simben@Opus.cabrillo.edu Wed Apr 20 10:47:01 2011
Date: Wed, 20 Apr 2011 10:47:01 -0700
From: Benji Simms <simben@Opus.cabrillo.edu>
Subject: Output from your job 150
To: simben@Opus.cabrillo.edu
```

*Because, you may not be online when the command runs, any error messages are mailed to you.*

```
/bin/bash: line 2: kitty: command not found
```

## Viewing an at jobs

```
/home/cis90/simben $ atq
882      2014-11-03 15:05 a simben90
881      2014-11-03 15:00 a simben90
883      2014-11-07 01:00 a simben90
```

```
/home/cis90/simben $ at -c 883
```

```
#!/bin/sh
# atrun uid=1201 gid=190
# mail simben90 0
umask 2
HOSTNAME=cablab.cis.cabrillo.edu; export HOSTNAME
SELINUX_ROLE_REQUESTED=; export SELINUX_ROLE_REQUESTED
SHELL=/bin/bash; export SHELL
HISTSIZE=1000; export HISTSIZE
SSH_CLIENT=2601:9:6680:53b:8d5f:4722:4af4:186e\ 59885\ 2220; export SSH_CLIENT
SELINUX_USE_CURRENT_RANGE=; export SELINUX_USE_CURRENT_RANGE
QTDIR=/usr/lib/qt-3.3; export QTDIR
QTINC=/usr/lib/qt-3.3/include; export QTINC
SSH_TTY=/dev/pts/2; export SSH_TTY
USER=simben90; export USER
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:or=40;31:01:mi=01;05;37;41:sg=30;42:ca=30;41:tw=30;42:ow=34;42:st=37;44;
ex=01;32;*.tar=01;31;*.tgz=01;31;*.arj=01;31;*.taz=01;31;*.lzh=01;31;*.lza=01;31;*.lzx=01;31;*.zip=01;31;*.z=01;31;*.d=01;31;*.gz=01;31;*.i
z=01;31;*.xz=01;31;*.bz2=01;31;*.tbz2=01;31;*.bz=01;31;*.deb=01;31;*.rpm=01;31;*.jar=01;31;*.rar=01;31;*.ace=01;31;*.zoo=01;31;*.cpio=01;31;
*.7z=01;31;*.xz=01;31;*.jpg=01;35;*.jpeg=01;35;*.gif=01;35;*.bmp=01;35;*.pbm=01;35;*.pgm=01;35;*.ppm=01;35;*.tga=01;35;*.xpm=01;35;*.xpm=01;35;*.tif=01;35;*.tiff=01;
35;*.png=01;35;*.svg=01;35;*.svgz=01;35;*.mng=01;35;*.pcx=01;35;*.mov=01;35;*.mpg=01;35;*.mpeg=01;35;*.m2v=01;35;*.mkv=01;35;*.ogm=01;35;*.mp4=01;35;*.m4v=01;35;*.
mp4=01;35;*.vob=01;35;*.qt=01;35;*.nuv=01;35;*.wmv=01;35;*.asf=01;35;*.rm=01;35;*.rmvb=01;35;*.flc=01;35;*.avi=01;35;*.flv=01;35;*.gl=01;35;*.d1=01;35;
*.act=01;35;*.xwd=01;35;*.yuv=01;35;*.cgm=01;35;*.emf=01;35;*.ans=01;35;*.ogv=01;35;*.ogv=01;35;*.aac=01;36;*.aif=01;36;*.mid=01;36;*.midi=0
1;36;*.mka=01;36;*.mp3=01;36;*.mpc=01;36;*.ogg=01;36;*.ra=01;36;*.wav=01;36;*.axa=01;36;*.oga=01;36;*.spx=01;36;*.xspf=01;36; export LS_COLORS
USERNAME=; export USERNAME
MAIL=/var/spool/mail/simben90; export MAIL
PATH=/usr/lib/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/home/cis90/simben/..:/bin:/home/cis90/simben/bin:; export PATH
PWD=/home/cis90/simben; export PWD
LANG=en_US.UTF-8; export LANG
SELINUX_LEVEL_REQUESTED=; export SELINUX_LEVEL_REQUESTED
HISTCONTROL=ignoredups; export HISTCONTROL
SHLVL=1; export SHLVL
HOME=/home/cis90/simben; export HOME
BASH_ENV=/home/cis90/simben/.bashrc; export BASH_ENV
LOGNAME=simben90; export LOGNAME
QTLIB=/usr/lib/qt-3.3/lib; export QTLIB
CVS_RSH=ssh; export CVS_RSH
SSH_CONNECTION=2601:9:6680:53b:8d5f:4722:4af4:186e\ 59885\ 2607:f380:80f:f425::230\ 2220; export SSH_CONNECTION
LESSOPEN=|/usr/bin/lesspipe.sh %s; export LESSOPEN
G_BROKEN_FILENAMES=1; export G_BROKEN_FILENAMES
cd /home/cis90/simben || {
    echo 'Execution directory inaccessible' >&2
    exit 1
}
```

Use the -c option to view the contents of an at job

reduced in size to fit on slide

All these environment variables must be set to appropriate values so your commands since you may be no longer logged in

```
${SHELL:-/bin/sh} << 'marcinDELIMITER7acf33a1'
```

```
chmod 640 letter
```

This is where you will see your own commands

```
marcinDELIMITER7acf33a1
/home/cis90/simben $
```



## Schedule a backup

### You try it:

```
/home/cis90/simben $ at now + 2 minutes
at> cp letter letter.bak
at> <EOT>
job 2481 at Tue Apr 9 15:09:00 2019
/home/cis90/simben $ atq
2481 Tue Apr 9 15:09:00 2019 a simben90
/home/cis90/simben $ at -c 2481 | tail
< snipped >
${SHELL:-/bin/sh} <<
'marcinDELIMITER026a792d'
cp letter letter.bak

marcinDELIMITER026a792d
/home/cis90/simben $ date
Tue Apr 9 15:08:06 PDT 2019
/home/cis90/simben $ ls letter*
letter
/home/cis90/simben $ date
Tue Apr 9 15:09:05 PDT 2019
/home/cis90/simben $ ls letter*
letter letter.bak
```

### You try it:

```
at now + 2 minutes
at> cp letter letter.bak
at> Ctrl-D
atq
at -c 2481 | tail
date
ls letter*
date
ls letter*
```

*Did it work?*

*Put your answer in the chat window.*

## Where does output go?

You try it:

```
/home/cis90/simben $ at now + 1 minute
```

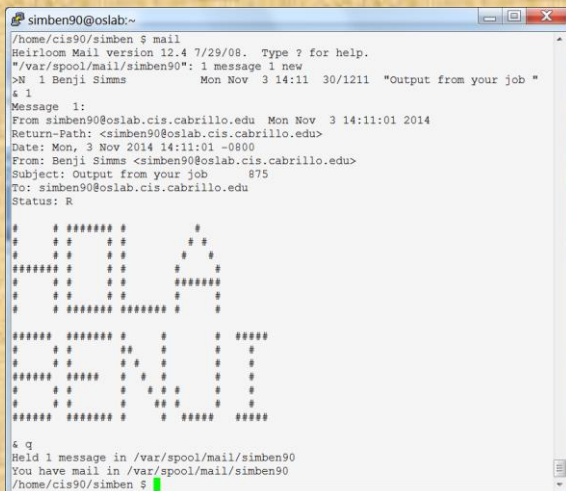
```
at> banner Hola Benji
```

```
at> <EOT> ← Use Ctrl-D for End of File
```

```
job 875 at 2014-11-03 14:11
```

```
/home/cis90/simben $ mail
```

Then read your mail a minute later



```
simben90@oslab:~
/home/cis90/simben $ mail
Beirloom Mail version 12.4 7/29/08. Type ? for help.
"/var/spool/mail/simben90": 1 message 1 new
>N 1 Benji Simms      Mon Nov  3 14:11 30/1211 "Output from your job"
4 1
Message 1:
From: simben90@oslab.cis.cabrillo.edu Mon Nov  3 14:11:01 2014
Return-Path: <simben90@oslab.cis.cabrillo.edu>
Date: Mon, 3 Nov 2014 14:11:01 -0800
From: Benji Simms <simben90@oslab.cis.cabrillo.edu>
Subject: Output from your job      875
To: simben90@oslab.cis.cabrillo.edu
Status: R

#  #####
#  #
#  #
##### #  #
#  #
#  #
#  #
#####
#  #
#  #
#  #
#####
#  #
#  #
#  #
#####

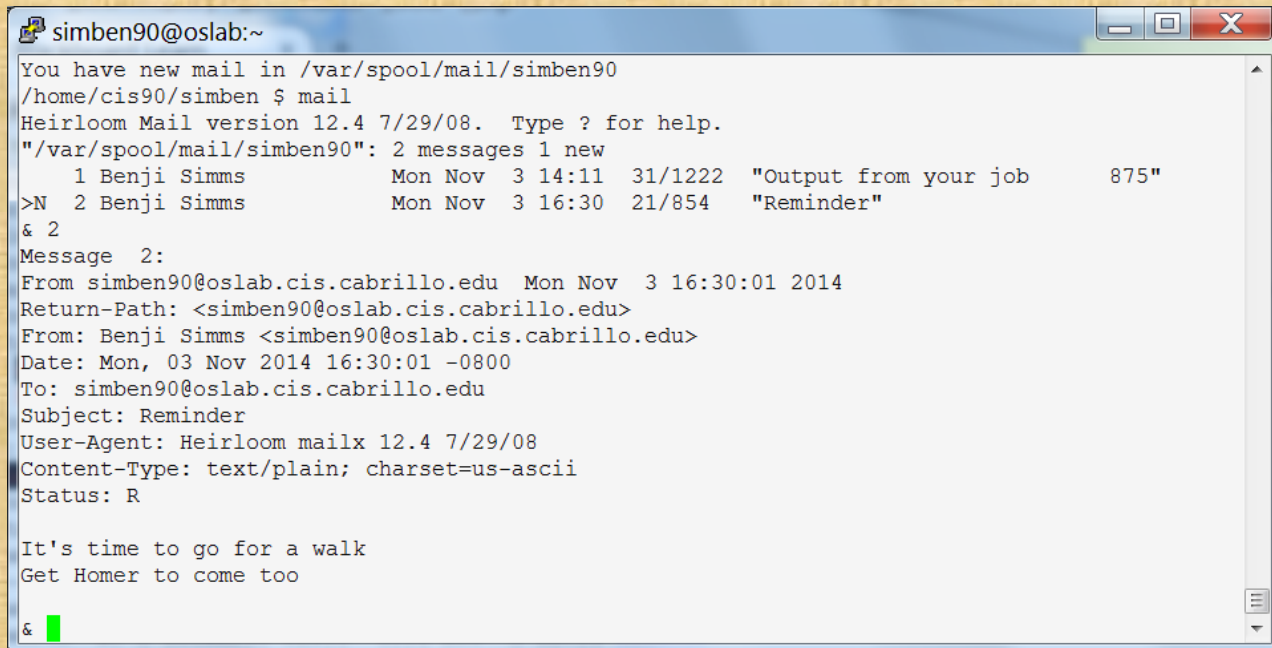
4 q
Read 1 message in /var/spool/mail/simben90
You have mail in /var/spool/mail/simben90
/home/cis90/simben $
```

*Write in the chat window the name of the sender of the email sent to you*

# Schedule an email

## Schedule an email reminder

```
/home/cis90/simben $ at 16:30
at> echo "It's time to go for a walk" > message
at> echo "Get Homer to come too" >> message
at> cat message | mail -s "Reminder" $LOGNAME
at> rm message
at> <EOT> Use Ctrl-D for End of File
/home/cis90/simben $
```



```
simben90@oslab:~
You have new mail in /var/spool/mail/simben90
/home/cis90/simben $ mail
Heirloom Mail version 12.4 7/29/08. Type ? for help.
"/var/spool/mail/simben90": 2 messages 1 new
 1 Benji Simms      Mon Nov  3 14:11  31/1222  "Output from your job      875"
>N 2 Benji Simms      Mon Nov  3 16:30  21/854   "Reminder"
& 2
Message 2:
From: simben90@oslab.cis.cabrillo.edu Mon Nov  3 16:30:01 2014
Return-Path: <simben90@oslab.cis.cabrillo.edu>
From: Benji Simms <simben90@oslab.cis.cabrillo.edu>
Date: Mon, 03 Nov 2014 16:30:01 -0800
To: simben90@oslab.cis.cabrillo.edu
Subject: Reminder
User-Agent: Heirloom mailx 12.4 7/29/08
Content-Type: text/plain; charset=us-ascii
Status: R

It's time to go for a walk
Get Homer to come too

&
```

## You try it:

```

simben90@oslab:~
/home/cis90/simben $
#           # ##### #           #
#           # #       # #       #
#           # #       # #       #
##### #           # #       #
#           # #       # #       #
#           # #       # #       #
#           # ##### #           #

##### # ##### #           # #       #
#           # #       # #       # #       #
#           # #       # #       # #       #
##### # ##### #           # #       #
#           # #       # #       # #       #
#           # #       # #       # #       #
##### # ##### #           # #       #

```

Write in the chat window the reason for doing a echo command before the banner command when writing to the terminal device

# text editors

## There are lots of text editors ...

### Windows

notepad  
notepad++  
textpad

*Text editors and word processors are different!*

### Mac

TextWrangler

- *Word processors are used by many different people to create documents containing text and graphics.*

### Linux

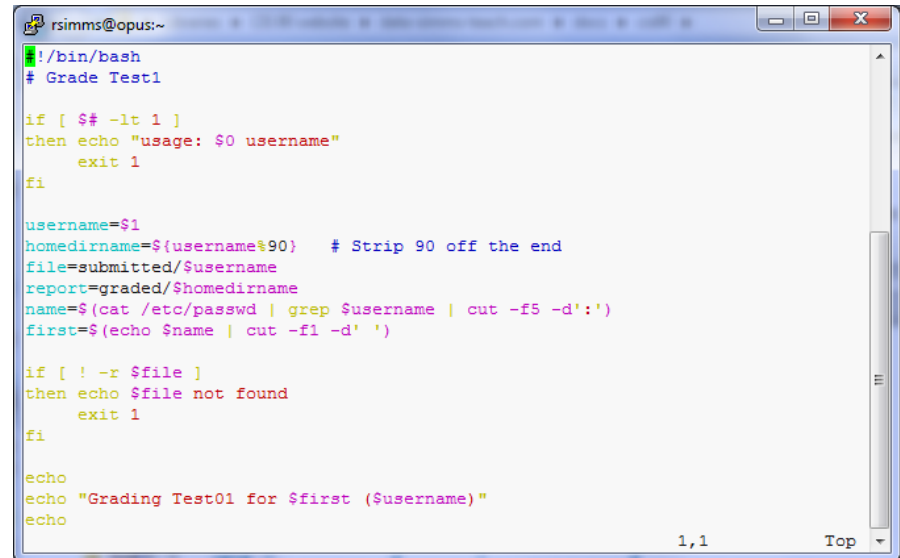
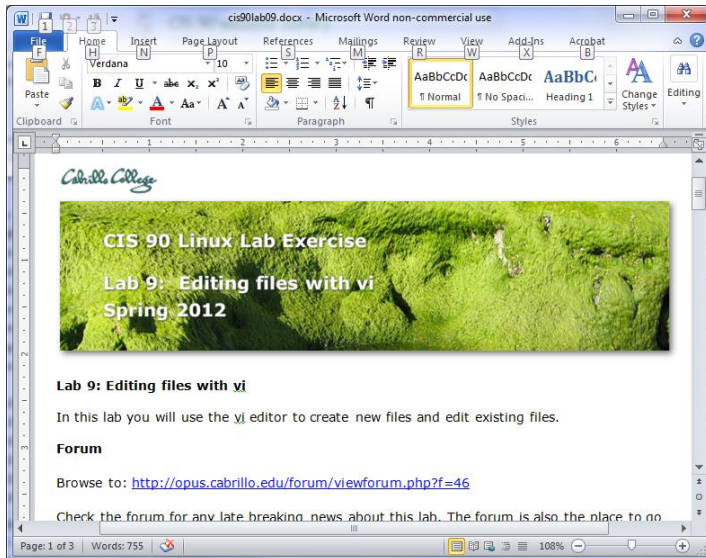
gedit  
emacs  
nano  
vi  
jove

- *Text editors are used by programmers to develop software and web designers to create web sites.*



*Thanks Maria!*





**Word processors** allow a rich set of formatting (fonts, sizes, styles, color) and graphics to be added to documents.

**Text editors** use color to show the language syntax

**vi 101**

## On Opus-II we are actually running VIM

```
/home/cis90/simben $ type -a vi  
vi is aliased to `vim'  
vi is /bin/vi  
/home/cis90/simben $ type vim  
vim is hashed (/usr/bin/vim)
```

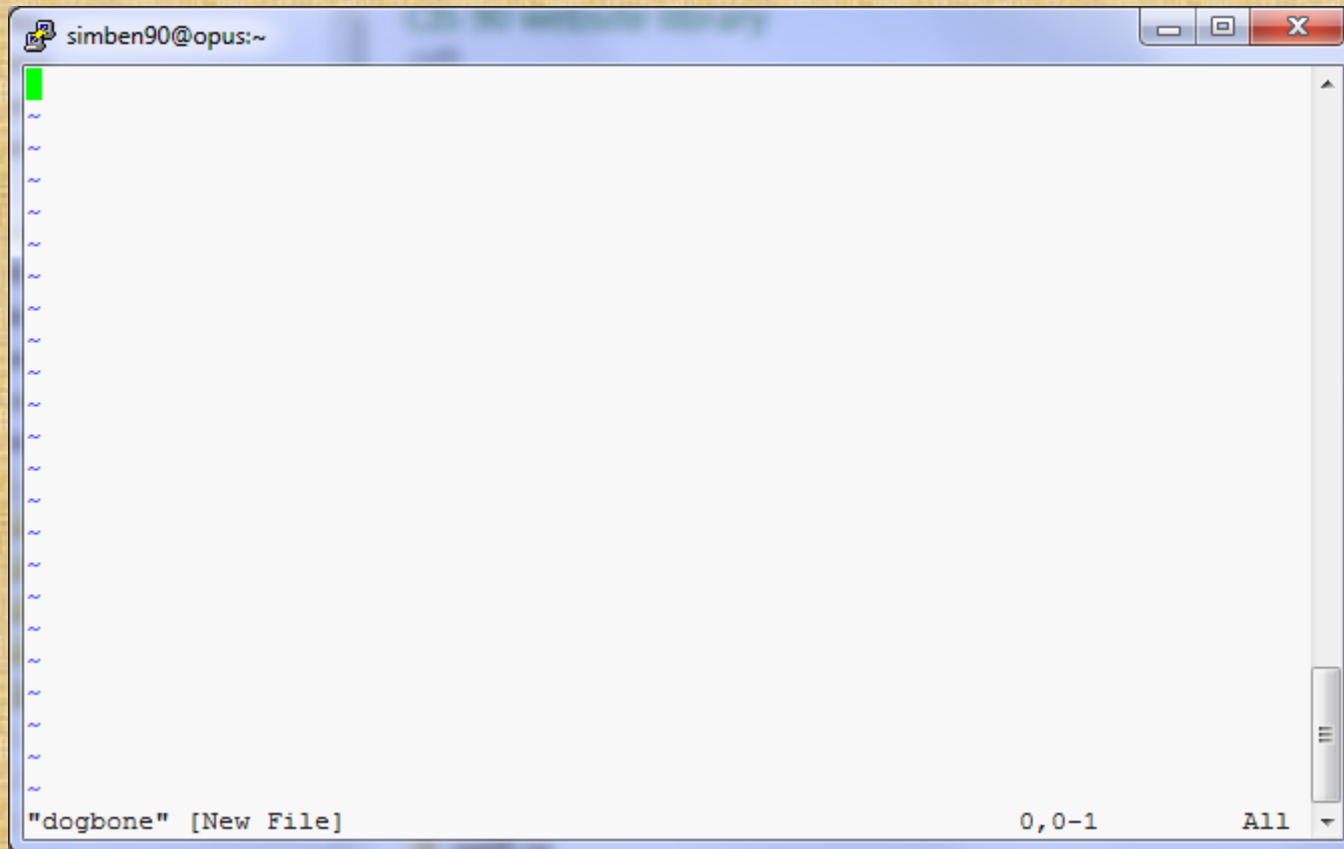
### History:

- The original vi code was written by Bill Joy for BSD Unix
- Bill Joy co-founded Sun Microsystems in 1982
- vi (for "visual")
- vim is an enhanced version of vi

```
/home/cis90/simben $ vi dogbone
```

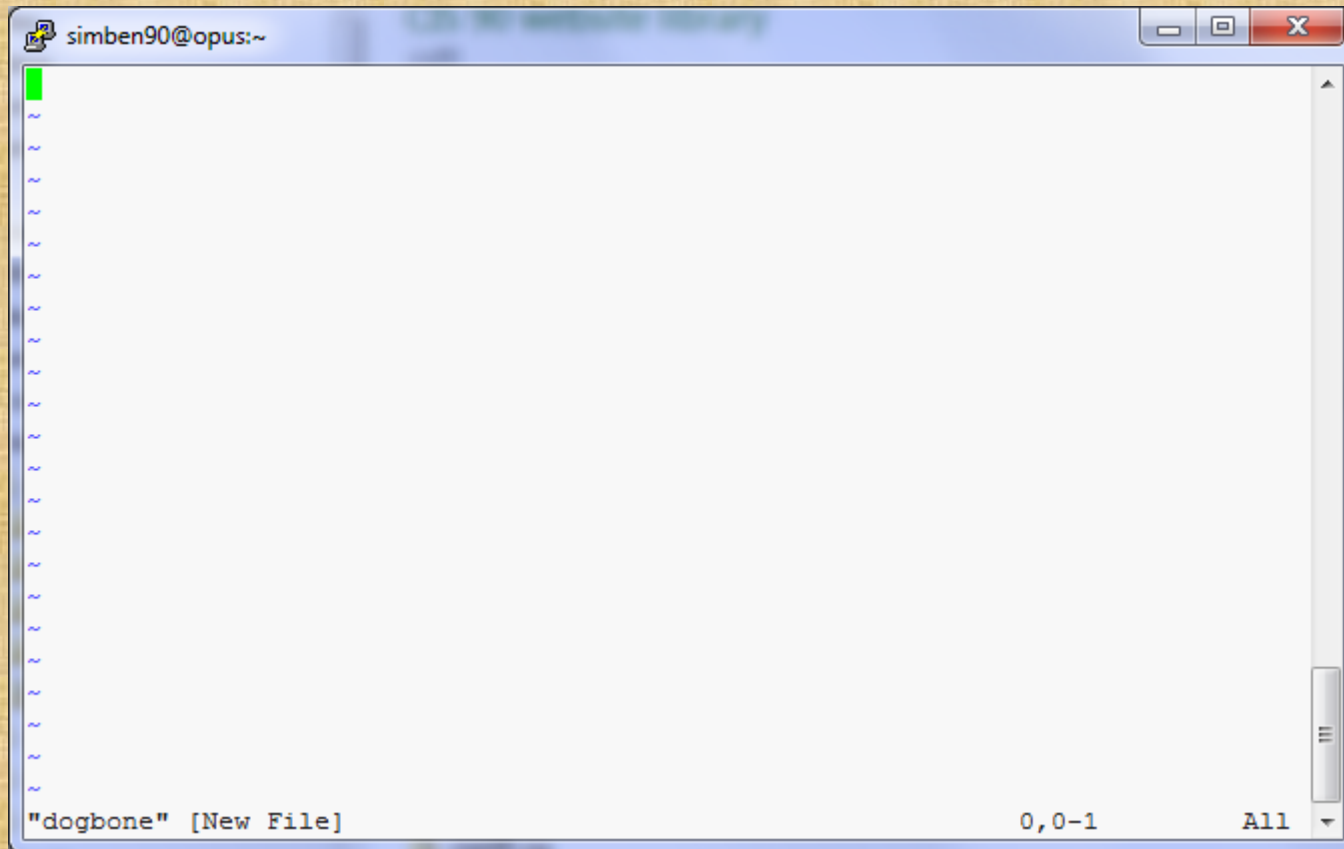
*Type this*

*See this ...*



**Take your hands OFF THE MOUSE – don't use it in vi!**

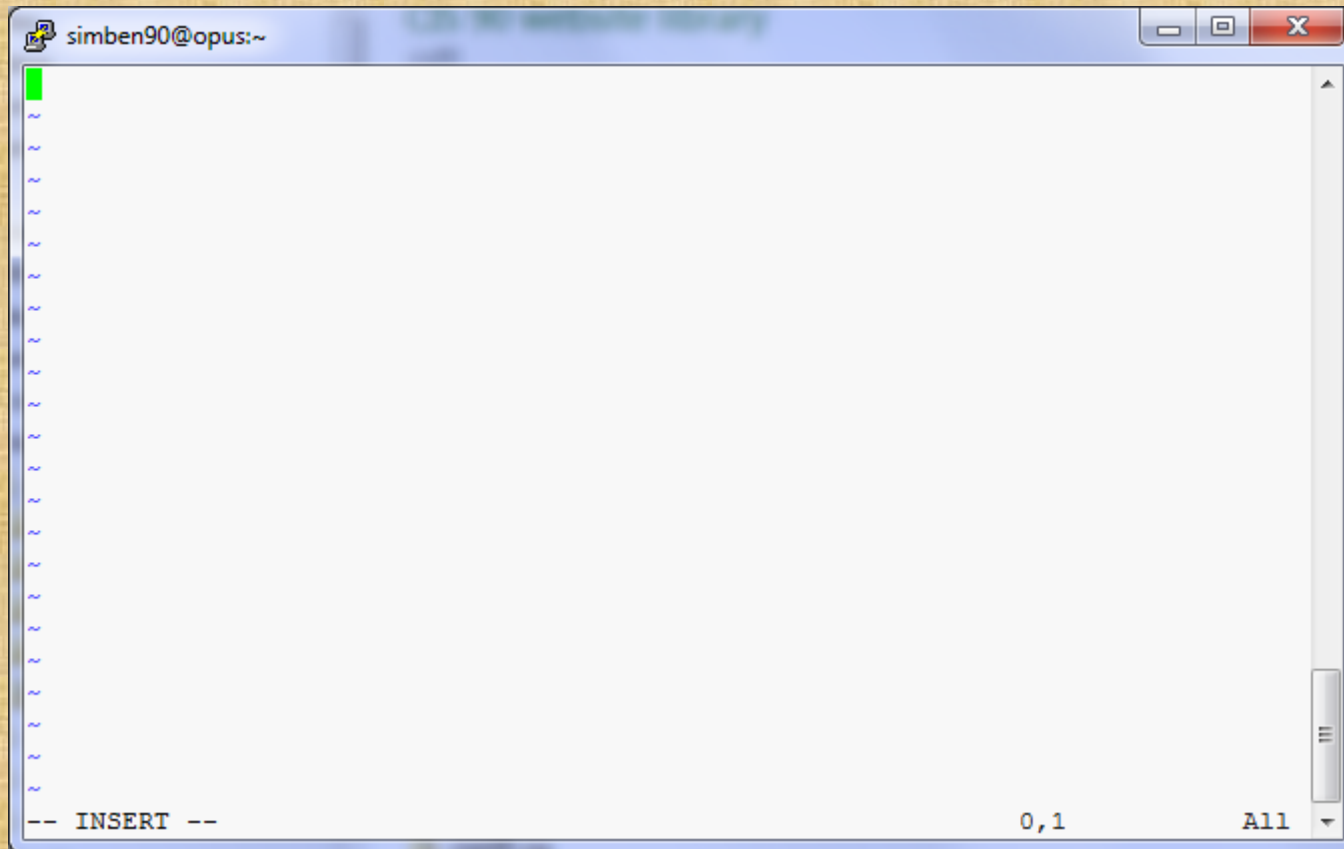
***Tap the letter i key (for insert)***



**Take your hands OFF THE MOUSE – don't use it in vi!**



***See this ...***



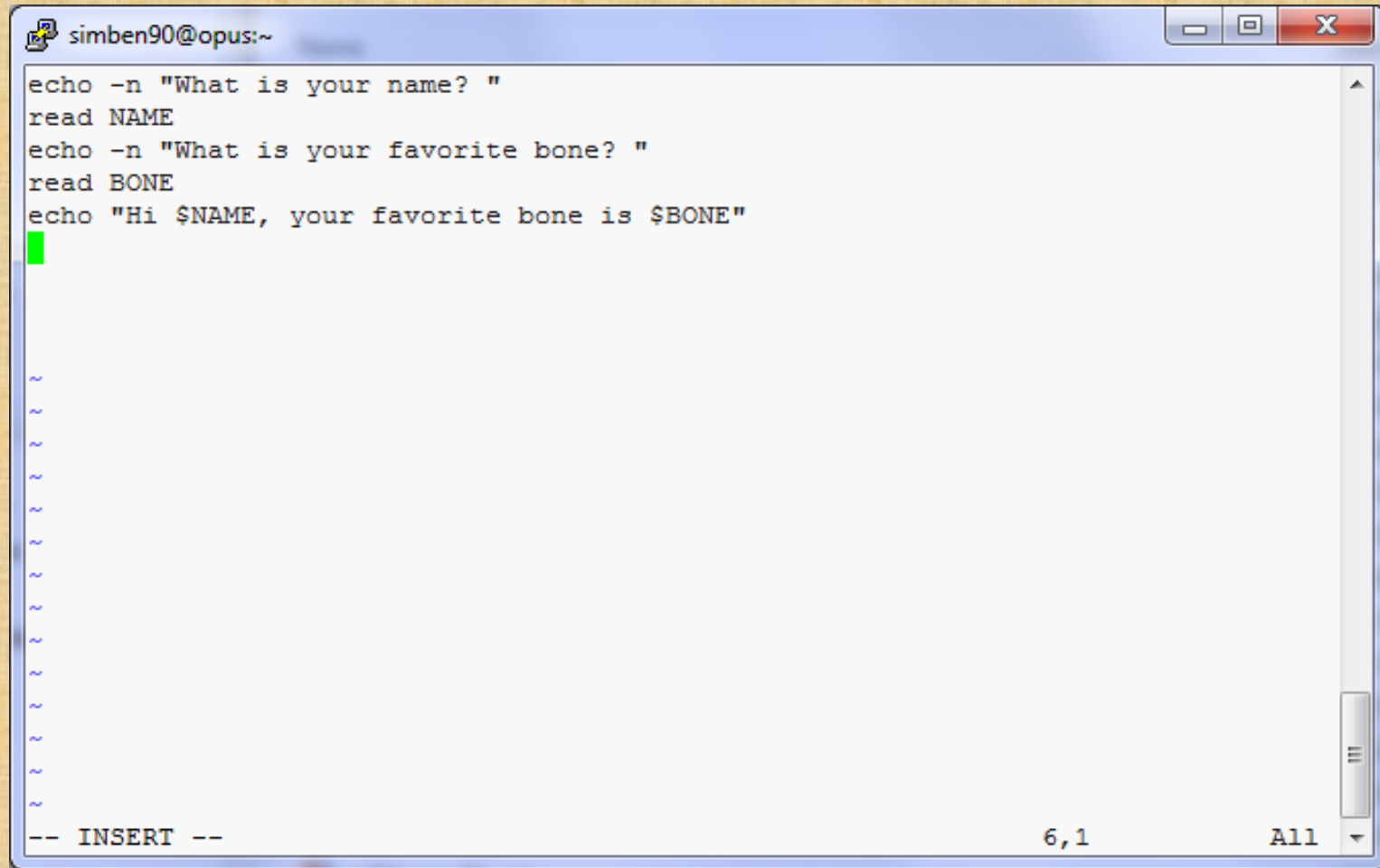
**Take your hands OFF THE MOUSE – don't use it in vi!**

***Very carefully type these five lines***

```
simben90@opus:~  
echo -n "What is your name? "  
read NAME  
echo -n "What is your favorite bone? "  
read BONE  
echo "Hi $NAME, your favorite bone is $BONE"  
█
```

**Take your hands OFF THE MOUSE – don't use it in vi!**

***Have your neighbor check that your five lines are PERFECT***



The screenshot shows a terminal window titled "simben90@opus:~". Inside the terminal, a shell script is being edited using the vi editor. The script contains the following lines:

```
echo -n "What is your name? "  
read NAME  
echo -n "What is your favorite bone? "  
read BONE  
echo "Hi $NAME, your favorite bone is $BONE"
```

The cursor is positioned at the end of the fifth line. The vi editor's status bar at the bottom indicates "-- INSERT --", the current cursor position is "6,1", and the file name is "All".

**Take your hands OFF THE MOUSE – don't use it in vi!**

## Tap the esc key

The screenshot shows a terminal window titled "simben90@opus:~". The terminal contains the following commands and output:

```
echo -n "What is your name? "  
read NAME  
echo -n "What is your favorite bone? "  
read BONE  
echo "Hi $NAME, your favorite bone is $BONE"
```

A green cursor is visible on the line following the last command. Below the main code block, there are several tilde (~) symbols representing additional lines of code or output that have been scrolled out of view. In the bottom right corner of the terminal window, the status bar displays "6,0-1" and "All".

## Take your hands OFF THE MOUSE – don't use it in vi!

A screenshot of a terminal window titled "simben90@opus:~". The window contains a shell script with three lines: `echo -n "What is your name? "`, `read NAME`, `echo -n "What is your favorite bone? "`, `read BONE`, and `echo "Hi $NAME, your favorite bone is $BONE"`. Below the script, there are several tilde (~) characters representing input or output, followed by a green cursor at the bottom left. The terminal has standard window controls (minimize, maximize, close) in the top right corner.

## Take your hands OFF THE MOUSE – don't use it in vi!

**Type wq**

A screenshot of a terminal window titled "simben90@opus:~". The window contains a shell script with four lines:  
`echo -n "What is your name? "  
read NAME  
echo -n "What is your favorite bone? "  
read BONE  
echo "Hi $NAME, your favorite bone is $BONE"`  
Below the script, there are several tilde (~) characters representing input or output. At the bottom left, the prompt ":wq" is visible next to a green cursor. The terminal has standard window controls (minimize, maximize, close) in the top right corner.

## Take your hands OFF THE MOUSE – don't use it in vi!



***Tap the enter key and see ...***

```
/home/cis90/simben $ vi dogbone  
/home/cis90/simben $
```

***Add execute permissions and try your new script***

```
/home/cis90/simben $ chmod +x dogbone
```

```
/home/cis90/simben $ dogbone
```

```
What is your name? Benji
```

```
What is your favorite bone? chicken
```

```
Hi Benji, your favorite bone is chicken
```

```
/home/cis90/simben $
```

# vi

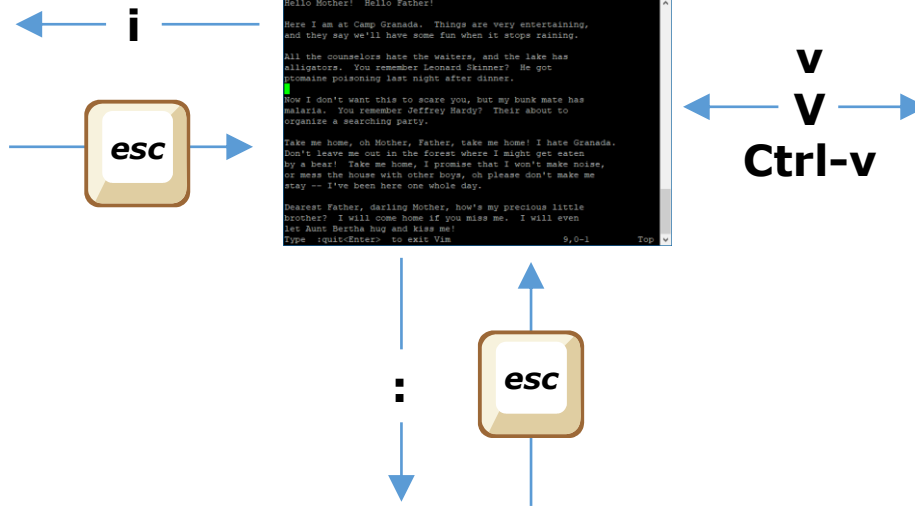
COMMAND mode  
INSERT mode  
command LINE mode

```
/home/cis90/simben $ cp letter myletter
/home/cis90/simben $ vi myletter
```

-- INSERT -- mode

COMMAND mode

-- VISUAL -- mode



Command LINE mode



## vi

### Moving around in a file

#### *Use in COMMAND mode*

**h** moves the cursor one character to the left  
**j** moves the cursor down one line  
**k** moves the cursor up one line  
**l** moves the cursor one character to the right

**^d** scrolls down 10 lines  
**^u** scrolls up 10 lines  
**^f** page forward one page  
**^b** page back one page

*Try typing a  
number in front of  
these commands  
and notice what  
happens*

*With vim (not vi) you can use arrow and  
page keys instead of these letter commands*

*Note: ^ is the Ctrl key*

## vi

### Moving around in a file

#### *Use in COMMAND mode*

**w** moves the cursor one "word" forward

**b** moves the cursor one "word" back

*Try typing a number in front  
of these commands and  
notice what happens*

**0** (zero) moves the cursor to the beginning of the line

**\$** moves the cursor to the end of the line

**G** moves the cursor to the last line in the file

**1G** moves the cursor to the first line in the file

**105G** moves the cursor to line 105



# vi

## Saving and Quitting

*Use in command LINE mode*

**:w** writes any changes to the file you are editing (like Save)

**:q** quits vi if you have saved your changes

**:q!** quits vi even if you haven't saved changes

**:wq** writes and quits

**:wq!** writes and quits vi even if you haven't saved changes

# vi

## Reading in and Writing out files

*Use in command LINE mode*

**:w filename** saves your file to a new name (like Save As)

**:w! filename** saves your file to a new name overwriting any previous data

**:r filename** reads in the contents of *filename* starting from the cursor position

**:e filename** replaces the current content with the content from *filename*

**:%s /string1/string2/g** replaces all string1 with string2 in the file

## vi

### Entering INSERT mode

*From COMMAND mode.*

- i** Ready to insert characters immediately before the current cursor position
- I** Ready to insert characters at the start of the current line
  
- a** Ready to append characters immediately after the current cursor position
- A** Ready to append characters at the end of the current line
  
- o** Ready to input characters in a new line that opens up below the cursor
- O** Ready to input characters in a new line that opens up above the cursor

## vi

### Cut, Copy, Pasting Commands

#### *Use in COMMAND mode*

**x** Deletes the current character

**r** Replace the current character with the character you type next

**dw** Deletes the current word

**dd** Deletes the current line

**D** Deletes to the end of the line

**yy** Copies a line to the clipboard buffer

**p** Pastes whatever is in the clipboard buffer below the current cursor

**P** Pastes whatever is in the clipboard buffer above the current cursor

## vi

### Miscellaneous Useful Commands

*Use in COMMAND mode.*

**^g** Tells you the filename you are editing and what line your cursor is on

**u** Undoes the last command you executed

**^r** Undo the undo (redo)

**.** Repeats the last command you executed

**/string** Searches for the string of characters in the file

**n** Finds the next occurrence of the current search string looking down the file

**N** Finds the next occurrence of the current search string looking up the file

**~** Changes the case of the current character

*Note: ^ is the Ctrl key*

## Use vi to edit your *edits/text.err* file

```
This is line number1.  
This is line number 1.  
Thi sis line line number 2.  
his is line number3.line number3.  
This is This is line #4.  
this number5 is line .  
Here is line number      6.  
This is lamw number      7.  
Thi is line number9.  
This is line  
number10.
```



```
This is line number 1.  
This is line number 2.  
This is line number 3.  
This is line number 4.  
This is line number 5.  
This is line number 6.  
This is line number 7.  
This is line number 8.  
This is line number 9.  
This is line number 10.
```

*Copy your corrected file into the chat window when finished*



[http://vim.wikia.com/wiki/Main\\_Page](http://vim.wikia.com/wiki/Main_Page)



*Tips and tricks for VIM users*

## The Mug of vi

The Mug of Vi - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://nostarch.com/mug.htm


Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Cabrillo College Home Page (0 unread) Yahoo! Mail, richsimms The Mug of Vi Sams Publishing - Contact Us

NO STARCH PRESS  
"the finest in geek entertainment"™

Home | Catalog | Where to buy | About | Jobs | Media | Blog | Cart

Google Custom Search Search



**The Mug of Vi**  
12 ounce heavy-duty ceramic

**\$12.95**

Order now

Hydration harmony

Copyright © 2000 No Starch Press

[See mug text](#)

Done

Big Mug Label - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://nostarch.com/mug\_big.htm

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Cabrillo College Home Page (0 unread) Yahoo! Mail, richsimms Big Mug Label Sams Publishing - Contact Us

NO STARCH PRESS  
"the finest in geek entertainment"™

Home | Catalog | Where to buy | About | Jobs | Media | Blog | Cart

Google Custom Search Search

Click on the image to return to **Mug of Vi** main page.

THE MUG OF VI		FILE COMMANDS		DELETING / INSERTING TEXT		MOVING AROUND		CUT / COPY / PASTE		WICKED COOL STUFF		
vi	filename(s)	edit a file or files	dw, dd, x	delete word, line, character	0	go to beginning of line (zero)	0	go to beginning of line (zero)	0	go to beginning of line (zero)	0	go to beginning of line (zero)
vi	-x filename	retrieve saved file after crash	ndd, nX	delete n lines, n characters	), {	move to next, previous sentence	), {	move to next, previous sentence	), {	move to next, previous sentence	), {	move to next, previous sentence
ZZ, :wq, :x		save and exit	x, X	delete character forward, backward	w, b	move forward, back one word	w, b	move forward, back one word	w, b	move forward, back one word	w, b	move forward, back one word
iq, :q!		quit; quit without saving	D, d\$	delete to end of line	e	go to end of current or next word	e	go to end of current or next word	e	go to end of current or next word	e	go to end of current or next word
:w, :wq, :w!	filename	save file, save file as filename	dmotion	delete from cursor to motion (\$, 0, etc.)								
:e filename		edit filename										
:r filename		insert filename	>, :<	indent, outdent line	yy, nY	copy n lines	yy, nY	copy n lines	yy, nY	copy n lines	yy, nY	copy n lines
:sh		drop to shell	S	replace text with blank line	yw, yy	copy word, line	yw, yy	copy word, line	yw, yy	copy word, line	yw, yy	copy word, line
:!cmd		run command cmd	o, O	insert new line below, above	p, P	paste text after, before cursor	p, P	paste text after, before cursor	p, P	paste text after, before cursor	p, P	paste text after, before cursor
:r !cmd		execute cmd and insert output	o, O	current line	a, i	insert text after, before cursor	a, i	insert text after, before cursor	a, i	insert text after, before cursor	a, i	insert text after, before cursor
/txt, ?txt		find txt forward or backward	u	undo last change	A, I	insert text end, beginning of line	A, I	insert text end, beginning of line	A, I	insert text end, beginning of line	A, I	insert text end, beginning of line
/*txt		find next line that starts with txt	~	repeat last change								
n, N		repeat last search backward, forward										
R		replace text from current character										

Done

## CafePress - VI Reference Mug

Amazon.com | CafePress

Secure | <https://www.amazon.com/CafePress-Reference-Mug-Unique-Coffee/dp/B0725PLKQ>

NEW & INTERESTING FINDS ON AMAZON EXPLORE

amazon Home & Kitchen

Deliver to Richard Soquel 95073

Departments Browsing History Richard's Amazon.com Today's Deals

EN Hello, Richard Account & Lists Orders Try Prime Cart

prime student 50% off Prime for College Students. LEARN MORE

Home & Kitchen > Kitchen & Dining > Dining & Entertaining > Novelty > Drinkware > Coffee Mugs

CafePress

CafePress - Vi Reference Mug - Unique Coffee Mug, Coffee Cup

★★★★☆ 3 customer reviews

Price: \$14.95 & FREE Shipping

Get \$50 off instantly: Pay \$0.00 upon approval for the Amazon Rewards Card

In Stock.

Get it as soon as April 18 - 23 when you choose Expedited Shipping. Ships from and sold by CafePress.

Size: Small

Small Mega

Color: White/Black Inside

- Small White & White / Black Coffee Mug is 3.75"x3", 11 oz capacity. Option is 4.75"x3.25", 20 oz capacity.
- Durable white ceramic with easy grip handle give a classic coffee cup feel.
- Coffee cup designs are professionally printed. Make someone smile.

Roll over image to zoom in

COLLEGE STUDENT? Start a 6-month trial of Prime Student at no cost. 50% off Prime after. prime student Close and Don't show me this again



## /bin/mail and vi

```
/home/cis90/simben $ mail milhom90
```

```
Subject: Good Bones
```

```
Hey Homer,
```

```
I really appreciate thatbone you sent me last week.
```

```
Let me knwo if you want to go mark some fench posts  
this weekend.
```

```
Later,
```

```
Ben
```

*You are composing a message and you spot some typos ...  
CRUD ... what can you do?*

## /bin/mail and vi

```
/home/cis90/simben $ mail milhom90
```

```
Subject: Good Bones
```

```
Hey Homer,
```

```
I really appreciate thatbone you sent me last week.
```

```
Let me knwo if you want to go mark some fench posts  
this weekend.
```

```
Later,
```

```
Ben
```

```
~v
```

*Well ... you could try the ~v command*

/bin/mail and vi

A screenshot of a terminal window titled "simben90@oslab:~". The terminal displays an email message with the following content:

I really appreciate that bone you sent me last week.  
Let me know if you want to go mark some fence posts  
this weekend.

Later,  
Benji

The email body is followed by several tilde (~) characters representing line wrapping. At the bottom of the terminal, a status bar indicates: `"/tmp/ReJZQRnV" 6L, 143C written`. The terminal window has standard Linux window controls (minimize, maximize, close) in the top right corner.

*The message is loaded into vi where changes or additions can be made. :wq is used to save and quit vi*

## /bin/mail and vi

```
/home/cis90/simben $ mail milhom90
Subject: Good Bones
Hey Homer,
I really appreciate thatbone you sent me last week.
Let me knwo if you want to go mark some fench posts
this weekend.
Later,
Ben
~v
(continue)
.
EOT
/home/cis90/simben $
```

*The earlier text with typos is still showing, however the corrected version is what is actually sent.*



## /bin/mail and vi

```
/home/cis90/milhom $ mail
Heirloom Mail version 12.4 7/29/08.  Type ? for help.
"/var/spool/mail/milhom90": 157 messages 5 new 155 unread
>N157 Benji Simms          Mon Nov 10 14:05  25/952  "Good Bones"
& 157
Message 157:
From simben90@oslab.cis.cabrillo.edu  Mon Nov 10 14:05:20 2014
Return-Path: <simben90@oslab.cis.cabrillo.edu>
From: Benji Simms <simben90@oslab.cis.cabrillo.edu>
Date: Mon, 10 Nov 2014 14:05:20 -0800
To: milhom90@oslab.cis.cabrillo.edu
Subject: Good Bones
User-Agent: Heirloom mailx 12.4 7/29/08
Content-Type: text/plain; charset=us-ascii
Status: R
```

```
Hey Homer,
I really appreciate that bone you sent me last week.
Let me know if you want to go mark some fence posts
this weekend.
Later,
Benji
```

*The message Homer reads has all the  
typos fixed.*

&

## Fix an email message before sending

```
/home/cis90/simben/edits $ mail rsimms
Subject: test of vi
sdkfjas;dflkjas;lkdfj
~v
(continue)
.
EOT
/home/cis90/simben/edits $
```

### *Once in vi:*

- *Use i to enter insert mode*
- *make changes*
- *save with <Esc>:wq*



# Assignment





*Instructor: remember to mail students the tech file!*



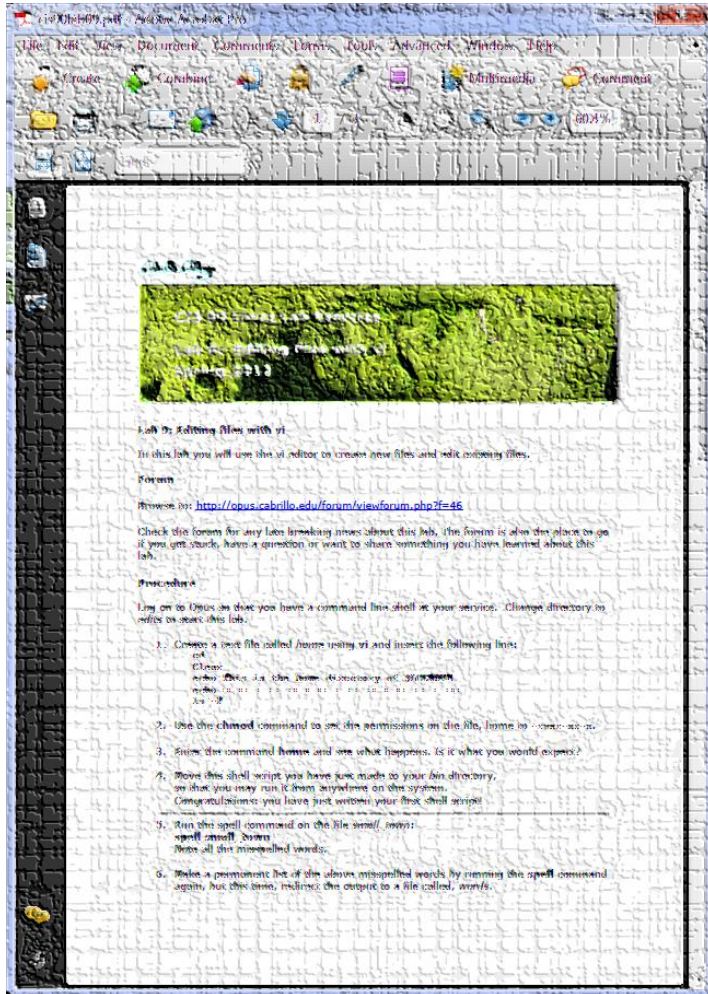
**~/cis90/lab09/mail-langs-all**

or

**at** *<end-of-class>*

**at> /home/rsimms/cis90/lab09/mail-langs-all**

**at> <Ctrl-D>**



Lab 9 will help  
you start building  
your vi skills!

A full-page background image showing a sunset over a beach. The sky is filled with vibrant orange, pink, and purple clouds. The sun is low on the horizon, casting a warm glow. To the right, a dark, silhouetted cliff rises from the beach. The foreground shows the wet sand of the beach reflecting the colors of the sky, with some dark rocks scattered about.

# Wrap up

New commands:

vi

Run vi editor

New Files and Directories:

na

na



## Next Class

Assignment: Check Calendar Page on web site to see what is due next week.

Lab 9  
Five Posts

Quiz questions for next class:

- How do you send a SIGKILL signal to one of your own processes?
- What vi command is used to exit vi without saving any of the changes you made?
- What vi commands are used for copy and paste?

# Backup

# The mystery of Ctrl-Z vs Ctrl-F

# Signals

## Special keystrokes

*Note: ^ is the Ctrl key*

```
/home/cis90/roddyduk$ stty -a
speed 38400 baud; rows 26; columns 78; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swch = <undef>; start = ^Q; stop = ^S; susp = ^F; rprnt = ^R;
werase = ^W; lnext = ^V; flush = ^O; min = 1; time = 0;
```

*Ctrl-f*

```
[rsimms@opus ~]$ stty -a
speed 38400 baud; rows 39; columns 84; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>; eol2 = <undef>;
swch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
```

*Ctrl-z*

*Why does the keystroke to send a Suspend (SIGTSTP or 20) signal differ between roddyduk (Ctrl-F) and rsimms (Ctrl-Z)?*

# Job Control

## A feature of the bash shell



Ctrl-Z or Ctrl-F (sends SIGTSTP 20 signal)

- Stops (suspends) a foreground process

```
[rsimms@opus ~]$ sleep 5
```

```
[1]+  Stopped                  sleep 5
```

*Ctrl-Z is tapped which  
stops the sleep  
command*

*PID 7728  
is stopped*

```
[rsimms@opus ~]$ ps -l -u rsimms
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
5	S	201	5368	5365	0	75	0	-	2460	-	?	00:00:00	sshd
0	S	201	5369	5368	0	76	0	-	1165	wait	pts/0	00:00:00	bash
5	S	201	6203	6200	0	75	0	-	2491	-	?	00:00:00	sshd
0	S	201	6204	6203	0	75	0	-	1165	-	pts/6	00:00:00	bash
0	T	201	7728	6204	0	75	0	-	926	finish	pts/6	00:00:00	sleep
0	R	201	7730	5369	0	78	0	-	1062	-	pts/0	00:00:00	ps

```
[rsimms@opus ~]$
```

# Job Control

## A feature of the bash shell

### **bg** command

- Resumes a suspended job in the background

```
[rsimms@opus ~]$ sleep 5
```

```
[1]+  Stopped                  sleep 5
```

```
[rsimms@opus ~]$ bg
```

```
[1]+ sleep 5 &
```

```
[rsimms@opus ~]$
```

*bg resumes the sleep command*

*PID 7728  
is gone*

```
[rsimms@opus ~]$ ps -l -u rsimms
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
5	S	201	5368	5365	0	75	0	-	2460	-	?	00:00:00	sshd
0	S	201	5369	5368	0	76	0	-	1165	wait	pts/0	00:00:00	bash
5	S	201	6203	6200	0	75	0	-	2491	-	?	00:00:00	sshd
0	S	201	6204	6203	0	75	0	-	1165	-	pts/6	00:00:00	bash
0	R	201	7742	5369	0	78	0	-	1061	-	pts/0	00:00:00	ps

```
[rsimms@opus ~]$
```



# Signals

## Jim's app script

```
rsimms@opus:/home/cis90/depot
#!/bin/sh
#
# app - script to demonstrate use of signals
#
# Usage:  run app with no options or parameters
#
# Send signals to it with keystrokes or kill command
#
# Notes:
# stty -echo stop the display of characters typed
# stty echo makes typed characters visible again
# stty susp ^Z sets suspend keystroke to Ctrl-Z (to stop foreground processes)
# stty susp @ sets suspend character to @ (to stop foreground processes)
#
trap '' 2 #Ignore SIGINT
trap 'echo -n quit it!' 3 #Handle SIGQUIT
trap 'stty echo susp ^Z;echo ee; echo cleanup;exit' 15 #Handle SIGTERM
clear
banner testing
stty -echo susp @
sleep 1
echo one
sleep 1
echo two
sleep 1
echo -n thr
while :
do sleep 1
done
~
```

*This is why Ctrl-F (suspend) stopped working and we had to use Ctrl-Z*

13,1 All



# Tangent on bg and SIGCONT

# Signals

*What is  
signal  
18?*



# Signals

SIGSTKFLT	16	Stack fault
SIGCHLD	17	Child process has stopped or exited, changed (POSIX)
SIGCONT	18	Continue executing, if stopped (POSIX)
SIGSTOP	19	Stop executing(can't be caught or ignored) (POSIX)
SIGTSTP	20	Terminal stop signal (POSIX) <b>Ctrl-Z or Ctrl-F</b>
SIGTTIN	21	Background process trying to read, from TTY (POSIX)
SIGTTOU	22	Background process trying to write, to TTY (POSIX)
SIGURG	23	Urgent condition on socket (4.2 BSD)
SIGXCPU	24	CPU limit exceeded (4.2 BSD)
SIGXFSZ	25	File size limit exceeded (4.2 BSD)
SIGVTALRM	26	Virtual alarm clock (4.2 BSD)
SIGPROF	27	Profiling alarm clock (4.2 BSD)
SIGWINCH	28	Window size change (4.3 BSD, Sun)
SIGIO	29	I/O now possible (4.2 BSD)
SIGPWR	30	Power failure restart (System V)

*Signal 18 continues a stopped process ... isn't that what bg does?*

The `bg` command is used to resume a stopped process

```
/home/cis90/roddyduk $ sleep 60
Ctrl-F (or Ctrl-Z) typed here
[1]+  Stopped                  sleep 60
/home/cis90/roddyduk $ bg
[1]+  sleep 60 &
/home/cis90/roddyduk $ jobs
[1]+  Running                  sleep 60 &
/home/cis90/roddyduk $ jobs
[1]+  Running                  sleep 60 &
/home/cis90/roddyduk $ jobs
[1]+  Done                     sleep 60
/home/cis90/roddyduk $
```

*bg resumed the stopped process which runs till it is finished*

*Instead of using **bg** to resume a stopped process in the background, lets try a **SIGCONT** (signal 18) instead*

```
/home/cis90/roddyduk $ sleep 60
```

Ctrl-F (or Ctrl-Z) typed here

```
[1]+  Stopped                  sleep 60
```

```
/home/cis90/roddyduk $ ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	10705	10704	0	76	0	-	1165	wait	pts/0	00:00:00	bash
0	T	1000	10743	10705	0	75	0	-	926	finish	pts/0	00:00:00	sleep
0	R	1000	10744	10705	0	78	0	-	1051	-	pts/0	00:00:00	ps

```
/home/cis90/roddyduk $ jobs
```

```
[1]+  Stopped                  sleep 60
```

```
/home/cis90/roddyduk $ kill -18 10743
```

```
/home/cis90/roddyduk $ jobs
```

```
[1]+  Running                  sleep 60 &
```

```
/home/cis90/roddyduk $ ps -l
```

F	S	UID	PID	PPID	C	PRI	NI	ADDR	SZ	WCHAN	TTY	TIME	CMD
0	S	1000	10705	10704	0	75	0	-	1165	wait	pts/0	00:00:00	bash
0	S	1000	10743	10705	0	85	0	-	926	322800	pts/0	00:00:00	sleep
0	R	1000	10746	10705	0	77	0	-	1050	-	pts/0	00:00:00	ps

```
/home/cis90/roddyduk $ jobs
```

```
[1]+  Running                  sleep 60 &
```

```
/home/cis90/roddyduk $ jobs
```

```
[1]+  Running                  sleep 60 &
```

```
/home/cis90/roddyduk $ jobs
```

```
[1]+  Done                     sleep 60
```

*Note sending a 18 signal or using the **bg** command will resume a stopped process*